Red Hat Satellite 5.6 Reference Guide

A guide for Red Hat Satellite advanced features
Edition 1

John Ha
Athene Chan
Lana Brindley
David O'Brien
Daniel Macpherson

Red Hat Satellite 5.6 Reference Guide

A guide for Red Hat Satellite advanced features
Edition 1

John Ha
Red Hat Engineering Content Services

Lana Brindley
Red Hat Engineering Content Services

Daniel Macpherson
Red Hat Engineering Content Services

Athene Chan
Red Hat Engineering Content Services

David O'Brien
Red Hat Engineering Content Services

Legal Notice

Copyright © 2013 Red Hat, Inc.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Welcome to the Red Hat Satellite 5.6 Reference Guide. The Red Hat Satellite Reference Guide guides you through advanced features of the Satellite server.

Table of Contents

Preface **7**
1. Audience 7
2. Document Conventions 7
2.1. Typographic Conventions 7
2.2. Pull-quote Conventions 8
2.3. Notes and Warnings 9
3. Getting Help and Giving Feedback 9
3.1. Do You Need Help? 9
3.2. We Need Feedback! 10

Chapter 1. Red Hat Satellite Information **11**
1.1. Command Line Configuration Management Tools 11
1.1.1. Red Hat Network Actions Control 11
1.1.1.1. General command line options 11
1.1.2. Red Hat Network Configuration Client 12
1.1.2.1. Listing Config Files 12
1.1.2.2. Getting a Config File 13
1.1.2.3. Viewing Config Channels 13
1.1.2.4. Differentiating between Config Files 13
1.1.2.5. Verifying Config Files 14
1.1.3. Red Hat Network Configuration Manager 14
1.1.3.1. Creating a Config Channel 15
1.1.3.2. Adding Files to a Config Channel 15
1.1.3.3. Differentiating between Latest Config Files 16
1.1.3.4. Differentiating between Various Versions 17
1.1.3.5. Downloading All Files in a Channel 17
1.1.3.6. Getting the Contents of a File 18
1.1.3.7. Listing All Files in a Channel 18
1.1.3.8. Listing All Config Channels 18
1.1.3.9. Removing a File from a Channel 19
1.1.3.10. Deleting a Config Channel 19
1.1.3.11. Determining the Number of File Revisions 19
1.1.3.12. Updating a File in a Channel 19
1.1.3.13. Uploading Multiple Files at Once 20
1.2. Monitoring 20
1.2.1. Prerequisites 21
1.2.2. Configuring the Red Hat Network Monitoring Daemon (rhnmd) 22
1.2.2.1. Installing the Red Hat Network Monitoring Daemon 22
1.2.2.2. Configuring SSH 23
1.2.2.3. Installing the SSH key 23
1.2.3. Configuring the mysql package for probes 24
1.2.4. Enabling Notifications 24
1.2.4.1. Creating Notification Methods 24
1.2.4.2. Receiving Notifications 25
1.2.4.3. Redirecting Notifications 25
1.2.4.4. Deleting Notification Methods 26
1.2.5. About Probes 27
1.2.5.1. Managing Probes 27
1.2.5.2. Establishing Thresholds 28
1.2.5.3. Monitoring the Satellite Server 28
1.2.6. Monitoring 28
1.2.6.1. Probe Status 28

1.2.6.1.1. Probe Status ⇒ Critical 29
1.2.6.1.2. Probe Status ⇒ Warning 29
1.2.6.1.3. Probe Status ⇒ Unknown 30
1.2.6.1.4. Probe Status ⇒ Pending 30
1.2.6.1.5. Probe Status ⇒ OK 30
1.2.6.1.6. Probe Status ⇒ All 30
1.2.6.1.7. Current State 30
1.2.6.2. Notification 31
1.2.6.2.1. Notification ⇒ Filters 31
1.2.6.2.1.1. Notification ⇒ Notification Filters ⇒ Active Filters 31
1.2.6.2.1.2. Notification ⇒ Notification Filters ⇒ Expired Filters 32
1.2.6.3. Probe Suites 32
1.2.6.4. Scout Config Push 34
1.2.6.5. General Monitoring Config 34
1.3. Multiple Satellites 34
1.3.1. Inter-Satellite Synchronization 35
1.3.1.1. Manual Configuration 35
1.3.1.2. Automated Configuration 37
1.3.2. Organizational Synchronization 39
1.3.3. Inter-Satellite Synchronization Use Cases 40

Chapter 2. Red Hat Satellite and Solaris-specific Information **42**
2.1. UNIX Support Guide 42
2.1.1. Introduction 42
2.1.1.1. Supported UNIX Variants 42
2.1.1.2. Prerequisites 42
2.1.1.3. Included Features 42
2.1.1.4. Differences in Functionality 43
2.1.1.5. Excluded Features 43
2.1.2. Satellite Server Preparation/Configuration 43
2.1.3. Unix Client System Preparation 45
2.1.3.1. Downloading and Installing Additional Packages 46
2.1.3.1.1. Install Third-Party Packages 46
2.1.3.1.2. Configuring the Library Search Path 47
2.1.3.1.3. Downloading Red Hat Network Client Packages 47
2.1.3.1.4. Installing the Red Hat Network Packages 48
2.1.3.1.5. Including Red Hat Network Packages in the PATH 48
2.1.3.2. Deploying Client SSL Certificates 48
2.1.3.3. Configuring the clients 49
2.1.4. Unix Client Registration and Updates 49
2.1.4.1. Registering Unix Systems 50
2.1.4.2. Obtaining Updates 50
2.1.4.2.1. Uploading Packages to the Satellite 51
2.1.4.2.1.1. solaris2mpm 51
2.1.4.2.1.2. rhnpush with .mpm Files 52
2.1.4.2.2. Updating Through the Website 52
2.1.4.2.3. rhnsd 52
2.1.4.2.4. Updating From the Command Line 53
2.1.5. Remote Commands 53
2.1.5.1. Enabling Commands 54
2.1.5.2. Issuing Commands 54

Chapter 3. Red Hat Satellite Proxy Information **55**
3.1. Using the Red Hat Network Package Manager and Serving Local Packages through the Red Hat Network Proxy 55
3.1.1. Creating a Private Channel 57

3.1.2. Uploading Packages 57

Chapter 4. Custom Package Management **59**
4.1. Building Packages for Red Hat Network 59
4.1.1. RPM Benefits 59
4.1.2. Red Hat Network RPM Guidelines 60
4.2. Digital Signatures for Red Hat Network Packages 61
4.2.1. Generating a GnuPG Keypair 61
4.2.2. Signing packages 63
4.3. Importing Custom GPG Keys 63

Chapter 5. Troubleshooting **65**
5.1. Disk Space 65
5.2. Installing and Updating 65
5.3. Services 66
5.4. Connectivity 66
5.5. Logging and Reporting 68
5.6. Errors 71
5.7. Web Interface 76
5.8. Anaconda 76
5.9. Tracebacks 78
5.10. Registration 79
5.11. Kickstarts and Snippets 80
5.12. Monitoring 81
5.13. Multi-Organization Satellites and Satellite Certificate 82
5.14. Proxy Installation and Configuration 83

Probes **89**
A.1. Probe Guidelines 89
A.2. Apache 1.3.x and 2.0.x 90
A.2.1. Apache::Processes 90
A.2.2. Apache::Traffic 91
A.2.3. Apache::Uptime 91
A.3. BEA WebLogic 6.x and higher 91
A.3.1. BEA WebLogic::Execute Queue 92
A.3.2. BEA WebLogic::Heap Free 93
A.3.3. BEA WebLogic::JDBC Connection Pool 93
A.3.4. BEA WebLogic::Server State 94
A.3.5. BEA WebLogic::Servlet 94
A.4. General 95
A.4.1. General::Remote Program 95
A.4.2. General::Remote Program with Data 95
A.4.3. General::SNMP Check 96
A.4.4. General::TCP Check 97
A.4.5. General::UDP Check 97
A.4.6. General::Uptime (SNMP) 98
A.5. Linux 98
A.5.1. Linux::CPU Usage 98
A.5.2. Linux::Disk IO Throughput 99
A.5.3. Linux::Disk Usage 99
A.5.4. Linux::Inodes 100
A.5.5. Linux::Interface Traffic 100
A.5.6. Linux::Load 101
A.5.7. Linux::Memory Usage 101
A.5.8. Linux::Process Counts by State 102
A.5.9. Linux::Process Count Total 102

A.5.10. Linux::Process Health 103
A.5.11. Linux::Process Running 104
A.5.12. Linux::Swap Usage 104
A.5.13. Linux::TCP Connections by State 104
A.5.14. Linux::Users 105
A.5.15. Linux::Virtual Memory 106
A.6. LogAgent 106
A.6.1. LogAgent::Log Pattern Match 106
A.6.2. LogAgent::Log Size 107
A.7. MySQL 3.23 - 3.33 108
A.7.1. MySQL::Database Accessibility 108
A.7.2. MySQL::Opened Tables 108
A.7.3. MySQL::Open Tables 109
A.7.4. MySQL::Query Rate 109
A.7.5. MySQL::Threads Running 110
A.8. Network Services 110
A.8.1. Network Services::DNS Lookup 110
A.8.2. Network Services::FTP 111
A.8.3. Network Services::IMAP Mail 111
A.8.4. Network Services::Mail Transfer (SMTP) 112
A.8.5. Network Services::Ping 112
A.8.6. Network Services::POP Mail 113
A.8.7. Network Services::Remote Ping 113
A.8.8. Network Services::RPCService 114
A.8.9. Network Services::Secure Web Server (HTTPS) 114
A.8.10. Network Services::SSH 115
A.8.11. Network Services::Web Server (HTTP) 115
A.9. Oracle 8i, 9i, 10g, and 11g 116
A.9.1. Oracle::Active Sessions 117
A.9.2. Oracle::Availability 117
A.9.3. Oracle::Blocking Sessions 117
A.9.4. Oracle::Buffer Cache 118
A.9.5. Oracle::Client Connectivity 118
A.9.6. Oracle::Data Dictionary Cache 119
A.9.7. Oracle::Disk Sort Ratio 119
A.9.8. Oracle::Idle Sessions 120
A.9.9. Oracle::Index Extents 120
A.9.10. Oracle::Library Cache 121
A.9.11. Oracle::Locks 121
A.9.12. Oracle::Redo Log 122
A.9.13. Oracle::Table Extents 122
A.9.14. Oracle::Tablespace Usage 123
A.9.15. Oracle::TNS Ping 124
A.10. Red Hat Satellite 124
A.10.1. Red Hat Satellite::Disk Space 124
A.10.2. Red Hat Satellite::Execution Time 125
A.10.3. Red Hat Satellite::Interface Traffic 125
A.10.4. Red Hat Satellite::Latency 125
A.10.5. Red Hat Satellite::Load 126
A.10.6. Red Hat Satellite::Probe Count 126
A.10.7. Red Hat Satellite::Process Counts 126
A.10.8. Red Hat Satellite::Processes 127
A.10.9. Red Hat Satellite::Process Health 127
A.10.10. Red Hat Satellite::Process Running 128
A.10.11. Red Hat Satellite::Swap 128

A.10.12. Red Hat Satellite::Users 128

Revision History **130**

Preface

1. Audience

The target audience for this guide includes **system administrators** who aim to manage updates for systems within an internal network.

2. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later include the Liberation Fonts set by default.

2.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

2.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                 struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

2.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

3. Getting Help and Giving Feedback

3.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer

Portal at http://access.redhat.com. Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products.
- submit a support case to Red Hat Global Support Services (GSS).
- access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at https://www.redhat.com/mailman/listinfo. Click on the name of any mailing list to subscribe to that list or to access the list archives.

3.2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **Red Hat Satellite 5.**

When submitting a bug report, be sure to mention the manual's identifier: *Docs Reference Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Red Hat Satellite Information

This section covers various topics on Red Hat Satellite advanced configuration.

1.1. Command Line Configuration Management Tools

In addition to the options provided in the Red Hat Satellite website, there are two command line tools for managing a system's configuration files: the **Red Hat Network Configuration Client** and the **Red Hat Network Configuration Manager**. There is a complementary **Red Hat Network Actions Control** tool that is used to enable and disable configuration management on client systems. If you do not yet have these these tools installed, they can be found within the **`Red Hat Network Tools`** child channel for your operating system.

Tip

Whenever a configuration file is deployed via the website, a backup of the previous file including its full path is made in the **`/var/lib/rhncfg/backups/`** directory on the affected system. The backup retains its filename but has a **`.rhn-cfg-backup`** extension appended.

1.1.1. Red Hat Network Actions Control

The **Red Hat Network Actions Control** (**`rhn-actions-control`**) application is used to enable and disable configuration management of a system. Client systems cannot be managed in this fashion by default. This tool allows System Administrators to enable or disable specific modes of allowable actions such as: *deploying* a configuration file onto the system, *uploading* a file from the system, using *diff* to find out what is currently managed on a system and what is available, or allowing running arbitrary *remote commands*. These various modes are enabled/disabled by placing/removing files and directories in the **`/etc/sysconfig/rhn/allowed-actions/`** directory. Due to the default permissions on the **`/etc/sysconfig/rhn/`** directory, Red Hat Network Actions Control have to be run by someone with root access.

1.1.1.1. General command line options

There is a **`man`** page available, as there are for most command line tools. Simply decide what Red Hat Network scheduled actions should be enabled for use by system administrators. These options enable the various scheduled action modes:

Table 1.1. `rhn-actions-control` options

Option	Description
--enable-deploy	Allow rhncfg-client to deploy files.
--enable-diff	Allow rhncfg-client to diff files.
--enable-upload	Allow rhncfg-client to upload files.
--enable-mtime-upload	Allow rhncfg-client to upload mtime.
--enable-all	Allow rhncfg-client to do everything.
--enable-run	Enable script.run
--disable-deploy	Disable deployment.
--disable-diff	Disable diff
--disable-upload	Disable upload
--disable-mtime-upload	Disable mtime upload
--disable-all	Disable all options
--disable-run	Disable script.run
--report	Report whether the modes are enabled or disabled
-f, --force	Force the operation without asking first
-h, --help	show help message and exit

Once a mode is set, your system is now ready for config management through Red Hat Satellite. **`rhn-actions-control --enable-all`** is a common option.

1.1.2. Red Hat Network Configuration Client

As the name implies, the **Red Hat Network Configuration Client** (**`rhncfg-client`**) is installed and run from an individual client system. From there you may use it to gain knowledge about how Red Hat Network deploys configuration files to the client.

The **Red Hat Network Configuration Client** offers these primary modes: list, get, channels, diff, and verify.

1.1.2.1. Listing Config Files

To list the configuration files for the machine and the labels of the config channels containing them, issue the command:

```
rhncfg-client list
```

The output resembles the following list:

```
Config Channel      File
config-channel-17   /etc/example-config.txt
config-channel-17   /var/spool/aalib.rpm
config-channel-14   /etc/rhn/rhn.conf
```

These are the configuration files that apply to your system. However, there may be duplicate files present in the other channels. For example, issue the following command:

```
rhncfg-manager list config-channel-14
```

and observe the following output:

```
Files in config channel 'config-channel-14' /etc/example-config.txt
/etc/rhn/rhn.conf
```

You may then wonder where the second version of **/etc/example-config.txt** went. The rank of the **/etc/example-config.txt** file in **config-channel-17** was higher than that of the same file in **config-channel-14**. As a result, the version of the configuration file in **config-channel-14** is not deployed for this system, although the file still resides in the channel. The **rhncfg-client** command does not list the file because it will not be deployed on this system.

1.1.2.2. Getting a Config File

To download the most relevant configuration file for the machine, issue the command:

```
rhncfg-client get /etc/example-config.txt
```

You should see output resembling:

```
Deploying /etc/example-config.txt
```

View the contents of the file with **less** or another pager. Note that the file is selected as the most relevant based upon the rank of the config channel containing it. This is accomplished within the **Configuration** tab of the **System Details** page.

1.1.2.3. Viewing Config Channels

To view the labels and names of the config channels that apply to the system, issue the command:

```
rhncfg-client channels
```

You should see output resembling:

```
Config channels: Label Name ----- ---- config-channel-17 config chan 2 config-
channel-14 config chan 1
```

The following table lists the options available for **rhncfg-client get**:

Table 1.2. rhncfg-client get options

Option	Description
--topdir=TOPDIR	Make all file operations relative to this string.
--exclude=EXCLUDE	Excludes a file from being deployed with 'get'/ May be used multiple times.
-h, --help	Show help message and exit

1.1.2.4. Differentiating between Config Files

To view the differences between the config files deployed on the system and those stored by Red Hat Network, issue the command:

```
rhncfg-client diff
```

The output resembles the following:

```
[root@testsatellite root]# rhncfg-client diff
--- /etc/test
+++ /etc/test 2013-08-28 00:14:49.405152824 +1000
@@ -1 +1,2 @@
 This is the first line
+This is the second line added
```

In addition, you may include the **--topdir** option to compare config files in Red Hat Network with those located in an arbitrary (and unused) location on the client system, like so:

```
[root@ root]# rhncfg-client diff --topdir /home/test/blah/ /usr/bin/diff:
/home/test/blah/etc/example-config.txt: No such file or directory /usr/bin/diff:
/home/test/blah/var/spool/aalib.rpm: No such file or directory
```

1.1.2.5. Verifying Config Files

To quickly determine if client configuration files are different than those associated with it via Red Hat Network, issue the command:

```
rhncfg-client verify
```

The output resembles the following:

```
modified /etc/example-config.txt /var/spool/aalib.rpm
```

The file **example-config.txt** is locally modified, while **aalib.rpm** is not.

The following table lists the options available for **rhncfg-client verify**:

Table 1.3. rhncfg-client verify options

Option	Description
-v, --verbose	Increase the amount of output detail. Displays differences in the mode, owner, and group permissions for the specified config file.
-o, --only	Only show files that differ.
-h, --help	Show help message and exit

1.1.3. Red Hat Network Configuration Manager

Unlike the **Red Hat Network Configuration Client**, the **Red Hat Network Configuration Manager** (**rhncfg-manager**) is designed to maintain Red Hat Network's central repository of config files and channels, not those located on client systems. This tool offers a command line alternative to the configuration management features within the Red Hat Network website, as well as the ability to script some or all of the related maintenance.

It is intended for use by Config Administrators and requires an Red Hat Network username and password that has the appropriate permission set. The username may be specified in **/etc/sysconfig/rhn/rhncfg-manager.conf** or in the [rhncfg-manager] section of **~/.rhncfgrc**.

When the **Red Hat Network Configuration Manager** is run as root, it attempts to pull in needed

configuration values from the **Red Hat Update Agent**. When run as a user other than root, you may have to make configuration changes within the **~/.rhncfgrc** file. The session file is cached in **~/.rhncfg-manager-session** to prevent logging in for every command.

The default timeout for the **Red Hat Network Configuration Manager** is 30 minutes. To alter this, add the **server.session_lifetime** option and new value to the **/etc/rhn/rhn.conf** file on the server running the manager, like so:

```
server.session_lifetime = 120
```

The **Red Hat Network Configuration Manager** offers these primary modes: add, create-channel, diff, diff-revisions, download-channel, get, list, list-channels, remove, remove-channel, revisions, update, and upload-channel.

Each mode offers its own set of options, which can be seen by issuing the following command:

```
rhncfg-manager mode --help
```

Replace ***mode*** with the name of the mode to be inspected:

```
rhncfg-manager diff-revisions --help
```

You can see such a list of options for the add mode at Table 1.4, "**rhncfg-manager add** options".

1.1.3.1. Creating a Config Channel

To create a config channel for your organization, issue the command:

```
rhncfg-manager create-channel channel-label
```

If prompted for your Red Hat Satellite username and password, provide them. The output resembles the following:

```
Red Hat Network username: rhn-user
Password:
Creating config channel channel-label Config channel channel-label created
```

Once you have created a config channel, use the remaining modes listed above to populate and maintain that channel.

1.1.3.2. Adding Files to a Config Channel

To add a file to a config channel, specify the channel label as well as the local file to be uploaded, such as:

```
rhncfg-manager add --channel=channel-label /path/to/file
```

In addition to the required channel label and the path to the file, you may use the available options for modifying the file during its addition. For instance, you may alter the path and file name by including the **--dest-file** option in the command, like:

```
rhncfg-manager add --channel=channel-label --dest-
file=/new/path/to/file.txt/path/to/file
```

The output resembles the following:

```
Pushing to channel example-channel
Local file >/path/to/file -> remote file /new/path/to/file.txt
```

The following table lists the options available for **`rhncfg-manager add`**:

Table 1.4. `rhncfg-manager add` options

Option	Description
-c CHANNEL --channel=CHANNEL	Upload files in this config channel
-d DEST_FILE --dest-file=DEST_FILE	Upload the file as this path
--delim-start=DELIM_START	Start delimiter for variable interpolation
--delim-end=DELIM_END	End delimiter for variable interpolation
-i, --ignore-missing	Ignore missing local files
--selinux-context=SELINUX_CONTEXT	Overwrite the SELinux context
-h, --help	show help message and exit

Note

By default, the maximum file size for configuration files is 128KB. If you need to change that value, find or create the following line in the **`/etc/rhn/rhn.conf`** file:

```
web.maximum_config_file_size=128
```

Additionally, find or create the following line in the **`/etc/rhn/rhn.conf`** file:

```
maximum_config_file_size=128
```

In both locations, change the value from 128 to whatever limit you want in bytes.

1.1.3.3. Differentiating between Latest Config Files

To view the differences between the config files on disk and the latest revisions in a channel, issue the command:

```
rhncfg-manager diff --channel=channel-label --dest-file=/path/to/file.txt \
/local/path/to/file
```

You should see output resembling:

```
--- /tmp/dest_path/example-config.txt config_channel: example-channel revision: 1
+++ /home/test/blah/hello_world.txt 2003-12-14 19:08:59.000000000 -0500
@@ -1 +1 @@
-foo
+hello, world
```

The following table lists the options available for **`rhncfg-manager diff`**:

Table 1.5. `rhncfg-manager diff` options

Option	Description
-c CHANNEL, --channel=CHANNEL	Get file(s) from this config channel
-r REVISION, --revision=REVISION	Use this revision
-d DEST_FILE, --dest-file=DEST_FILE	Upload the file as this path
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
-h, --help	Show help message and exit

1.1.3.4. Differentiating between Various Versions

To compare different versions of a file across channels and revisions, use the **-r** flag to indicate which revision of the file should be compared and the **-n** flag to identify the two channels to be checked. See Section 1.1.3.11, "Determining the Number of File Revisions" for related instructions. Specify only one file name here, since you are comparing the file against another version of itself. For example:

```
rhncfg-manager diff-revisions -n=channel-label1 -r=1 -n=channel-label2 -r=1
/path/to/file.txt
```

The output resembles the following:

```
--- /tmp/dest_path/example-config.txt 2004-01-13 14:36:41 \ config channel:
example-channel2 revision: 1
--- /tmp/dest_path/example-config.txt 2004-01-13 14:42:42 \ config channel:
example-channel3 revision: 1
@@ -1 +1,20 @@
-foo
+blah
+-----BEGIN PGP SIGNATURE-----
+Version: GnuPG v1.0.6 (GNU/Linux)
+Comment: For info see http://www.gnupg.org
+
+iD8DBQA9ZY6vse4XmfJPGwgRAsHcAJ9ud9dabUcdscdcqB8AZP7e0Fua0NmKsdhQCeOWHX
+VsDTfen2NWdwwPaTM+S+Cow=
+=Ltp2
+-----END PGP SIGNATURE-----
```

The following table lists the options available for **`rhncfg-manager diff-revisions`**:

Table 1.6. `rhncfg-manager diff-revisions` options

Option	Description
-c CHANNEL, --channel=CHANNEL	Use this config channel
-r REVISION, --revision=REVISION	Use this revision
-h, --help	Show help message and exit

1.1.3.5. Downloading All Files in a Channel

To download all the files in a channel to disk, create a directory and issue the following command:

```
rhncfg-manager download-channel channel-label --topdir .
```

The output resembles the following:

```
Copying /tmp/dest_path/example-config.txt -> \ blah2/tmp/dest_path/example-
config.txt
```

The following table lists the options available for **`rhncfg-manager download-channel`**:

Table 1.7. `rhncfg-manager download-channel` options

Option	Description
-t TOPDIR, --topdir=TOPDIR	Directory all the file paths are relative to. This option must be set.
-h, --help	Show help message and exit

1.1.3.6. Getting the Contents of a File

To direct the contents of a particular file to stdout, issue the command:

```
rhncfg-manager get --channel=channel-label \ /tmp/dest_path/example-config.txt
```

You should see the contents of the file as output.

1.1.3.7. Listing All Files in a Channel

To list all the files in a channel, issue the command:

```
rhncfg-manager list channel-label
```

You should see output resembling:

```
Files in config channel `example-channel3': /tmp/dest_path/example-config.txt
```

The following table lists the options available for **`rhncfg-manager get`**:

Table 1.8. `rhncfg-manager get` options

Option	Description
-c CHANNEL, --channel=CHANNEL	Get file(s) from this config channel
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
-r REVISION, --revision=REVISION	Get this file revision
-h, --help	Show help message and exit

1.1.3.8. Listing All Config Channels

To list all of your organization's configuration channels, issue the command:

```
rhncfg-manager list-channels
```

The output resembles the following:

```
Available config channels: example-channel example-channel2 example-channel3
config-channel-14 config-channel-17
```

Note that this does not list **local_override** or **server_import** channels.

1.1.3.9. Removing a File from a Channel

To remove a file from a channel, issue the command:

```
rhncfg-manager remove --channel=channel-label /tmp/dest_path/example-config.txt
```

If prompted for your Red Hat Network username and password, provide them. You should see output resembling:

```
Red Hat Network username: rhn-user Password: Removing from config channel example-
channel3 /tmp/dest_path/example-config.txt removed
```

The following table lists the options available for **rhncfg-manager remove**:

Table 1.9. rhncfg-manager remove options

Option	Description
-c CHANNEL, --channel=CHANNEL	Remove files from this config channel
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
-h, --help	Show help message and exit

1.1.3.10. Deleting a Config Channel

To destroy a config channel in your organization, issue the command:

```
rhncfg-manager remove-channel channel-label
```

The output resembles the following:

```
Removing config channel example-channel Config channel example-channel removed
```

1.1.3.11. Determining the Number of File Revisions

To find out how many revisions (revisions go from 1 to N where N is an integer greater than 0) of a file/path are in a channel, issue the following command:

```
rhncfg-manager revisions channel-label /tmp/dest_path/example-config.txt
```

The output resembles the following:

```
Analyzing files in config channel example-channel \ /tmp/dest_path/example-
config.txt: 1
```

1.1.3.12. Updating a File in a Channel

To create a new revision of a file in a channel (or add the first revision to that channel if none existed before for the given path), issue the following command:

```
rhncfg-manager update \ --channel=channel-label --dest-file=/path/to/file.txt
/local/path/to/file
```

The output resembles the following:

```
Pushing to channel example-channel: Local file example-
channel/tmp/dest_path/example-config.txt -> \ remote file /tmp/dest_path/example-
config.txt
```

The following table lists the options available for **`rhncfg-manager update`**:

Table 1.10. `rhncfg-manager update` options

Option	Description
-c CHANNEL, --channel=CHANNEL	Upload files in this config channel
-d DEST_FILE, --dest-file=DEST_FILE	Upload the file as this path
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
--delim-start=DELIM_START	Start delimiter for variable interpolation
--delim-end=DELIM_END	End delimiter for variable interpolation
-h, --help	Show help message and exit

1.1.3.13. Uploading Multiple Files at Once

To upload multiple files to a config channel from local disk at once, issue the command:

```
rhncfg-manager upload-channel --topdir=topdir channel-label
```

The output resembles the following:

```
Using config channel example-channel4 Uploading /tmp/ola_world.txt from
blah4/tmp/ola_world.txt
```

The following table lists the options available for **`rhncfg-manager upload-channel`**:

Table 1.11. `rhncfg-manager upload-channel` options

Option	Description
-t TOPDIR, --topdir=TOPDIR	Directory all the file paths are relative to
-c CHANNEL, --channel=CHANNEL	List of channels the config info will be uploaded into. Channels delimited by ','. Example: --channel=foo,bar,baz
-h, --help	Show help message and exit

1.2. Monitoring

The Red Hat Network Monitoring entitlement allows you to perform a whole host of actions designed to keep your systems running properly and efficiently. With it, you can keep close watch on system resources, network services, databases, and both standard and custom applications.

Monitoring provides both real time and historical state change information, as well as specific metric data. It provides notifications of system failures and performance degradation before it becomes critical. It also provides information that assists in capacity planning and event correlation. For example, the results of a probe recording CPU usage across systems would assist in balancing loads on those systems.

There are two components to the monitoring system: the monitoring system and the *monitoring scout*. The monitoring system is installed in the Satellite and performs backend functions such as storing monitoring data and acting on it. The monitoring scout runs all the probes and collects monitoring data. The monitoring scout can be enabled to run on a Satellite or Red Hat Satellite Proxy system. Using monitoring scout on Proxy allows you to offload work from the Satellite, providing scalability for probes.

Monitoring entails establishing notification methods, installing probes on systems, regularly reviewing the status of all probes, and generating reports displaying historical data for a system or service. This section seeks to identify common tasks associated with the Monitoring entitlement. Remember, virtually all changes affecting your Monitoring infrastructure must be finalized by updating your configuration, through the **`Scout Config Push`** page.

1.2.1. Prerequisites

Before attempting to implement Red Hat Network Monitoring within your infrastructure, ensure you have all of the necessary tools in place. At a minimum, you need:

- Monitoring entitlements - These entitlements are required for all systems that are to be monitored. Monitoring is only supported on Red Hat Enterprise Linux systems.
- Red Hat Satellite with monitoring - monitoring systems must be connected to a Satellite with a base operating system of Red Hat Enterprise Linux 5 or later.
- Monitoring Administrator - This role must be granted to users installing probes, creating notification methods, or altering the monitoring infrastructure in any way. (Remember, the Satellite Administrator automatically inherits the abilities of all other roles within an organization and can therefore conduct these tasks.). Assign this role through the **`User Details`** page for the user.
- Red Hat Network monitoring daemon - This daemon, along with the SSH key for the scout, is required on systems that are monitored in order for the internal process monitors to be executed. You may, however, be able to run these probes using the systems' existing SSH daemon (**`sshd`**). See Section 1.2.2, “Configuring the Red Hat Network Monitoring Daemon (**`rhnmd`**)” for installation instructions and a quick list of probes requiring this secure connection. See Appendix A, *Probes* for the complete list of available probes.

Enabling Monitoring

Monitoring is disabled by default, and will need to be enabled before it can be used.

1. Log in as a user with Satellite Administrator privileges and navigate to **Admin → Red Hat Satellite Configuration**. Click the **`Enable Monitoring`** checkbox, then click **`Update`** to save.
2. Restart services to pick up the changes. Go to the **`restart`** tab to restart the Satellite. This will take the Satellite offline for a few minutes.
3. Check if the **Monitoring** tab is available under **Red Hat Satellite Configuration** to confirm that monitoring is enabled.
4. Navigate to **Admin → Red Hat Satellite Configuration → Monitoring**. Click the **`Enable Monitoring Scout`** checkbox to enable the scout. Click **`Update Config`** to save.

Note

It is recommended that you leave the monitoring configuration values as the default values. **Sendmail** needs to be configured to use notifications.

1.2.2. Configuring the Red Hat Network Monitoring Daemon (rhnmd)

To make the most out of your monitoring entitlement, Red Hat suggests installing the Red Hat Network monitoring daemon on your client systems. Based upon **OpenSSH**, **rhnmd** enables the Satellite to communicate securely with the client system to access internal processes and retrieve probe status.

Please note that the Red Hat Network monitoring daemon requires that monitored systems allow connections on port 4545. You may avoid opening this port and installing the daemon altogether by using **sshd** instead. See Section 1.2.2.2, “Configuring SSH” for details.

Some probes require the daemon. An encrypted connection, either through the Red Hat Network monitoring daemon or **sshd**, is required on client systems for the following probes to run:

- Linux::CPU Usage
- Linux::Disk IO Throughput
- Linux::Disk Usage
- Linux::Inodes
- Linux::Interface Traffic
- Linux::Load
- Linux::Memory Usage
- Linux::Process Counts by State
- Linux::Process Count Total
- Linux::Process Health
- Linux::Process Running
- Linux::Swap Usage
- Linux::TCP Connections by State
- Linux::Users
- Linux::Virtual Memory
- LogAgent::Log Pattern Match
- LogAgent::Log Size
- Network Services::Remote Ping
- Oracle::Client Connectivity
- General::Remote Program
- General::Remote Program with Data

Note that all probes in the Linux group have this requirement.

1.2.2.1. Installing the Red Hat Network Monitoring Daemon

Install the Red Hat Network monitoring daemon to prepare systems for monitoring with the probes identified by **rhnmd**. Note that the steps in this section are optional if you intend to use **sshd** to allow secure connections between the Red Hat Network monitoring infrastructure and the monitored systems. See Section 1.2.2.2, “Configuring SSH” for instructions.

The **rhnmd** package can be found in the Red Hat Network Tools channel for all Red Hat Enterprise Linux distributions. To install it:

1. Subscribe the systems to be monitored to the Red Hat Network Tools channel associated with the system. This can be done individually through the **System Details** → **Channels** → **Software** subtab or for multiple systems at once through the **Channel Details** → **Target Systems** tab.
2. Once subscribed, open the **Channel Details** → **Packages** tab and find the **rhnmd** package (under 'R').
3. Click the package name to open the **Package Details** page. Go to the **Target Systems** tab, select the desired systems, and click **Install Packages**.
4. Install the SSH public key on all client systems to be monitored, as described in Section 1.2.2.3, "Installing the SSH key".
5. Start the Red Hat Network monitoring daemon on all client systems using the command:

```
service rhnmd start
```

6. When adding probes requiring the daemon, accept the default values for **RHNMD User** and **RHNMD Port**: **nocpulse** and **4545**, respectively.

1.2.2.2. Configuring SSH

If you wish to avoid installing the Red Hat Network monitoring daemon and opening port 4545 on client systems, you may configure **sshd** to provide the encrypted connection required between the systems and Red Hat Network. This may be especially desirable if you already have **sshd** running. To configure the daemon for monitoring use:

1. Ensure the SSH package is installed on the systems to be monitored:

```
rpm -qi openssh-server
```

2. Identify the user to be associated with the daemon. This can be any user available on the system, as long as the required SSH key can be put in the user's **~/.ssh/authorized_keys** file.
3. Identify the port used by the daemon, as identified in its **/etc/ssh/sshd_config** configuration file. The default is port 22.
4. Install the SSH public key on all client systems to be monitored, as described in Section 1.2.2.3, "Installing the SSH key".
5. Start the **sshd** on all client systems using the command:

```
service sshd start
```

6. When adding probes requiring the daemon, insert the values derived from steps 2 and 3 in the **RHNMD User** and **RHNMD Port** fields.

1.2.2.3. Installing the SSH key

Whether you use **rhnmd** or **sshd**, you must install the Red Hat Network monitoring daemon public SSH key on the systems to be monitored to complete the secure connection. To install it:

1. Navigate to the **Monitoring** → **Scout Config Push** page on the Satellite interface and click the name of the scout that will monitor the client system. The SSH **id_dsa.pub** key is visible on the resulting page.
2. Copy the character string (beginning with ssh-dss and ending with the hostname of the Satellite).
3. Select **Systems** from the left menu, and click the checkbox next to the systems you want to send

the SSH key to. Click the **Manage** button at the top to finish.

4. From the **System Set Manager**, click **Run remote commands**, then in the **Script** text box, type the following line:

```
#!/bin/sh
cat <<EOF >> ~nocpulse/.ssh/authorized_keys
```

Then, press **Enter**, paste the SSH Key and add EOF. The result should look similar to the following:

```
#!/bin/sh
cat <<EOF>> ~nocpulse/.ssh/authorized_keys
ssh-dss AABBAB3NzaC3kc3MABCCBAJ4cmyf5jt/ihdtFbNE1YHsT0np0SYJz7xk
hzoKUUWnZmOUqJ7eXoTbGEcZjZLppOZgzAepw1vUHXfa/L9XiXvsV8K5Qmcu70h0
1gohBIder/1I1QbHMCgfDVFPtfV5eedau4AAACAc99dHbWhk/dMPiWXgHxdI0vT2
SnuozIox2klmfbTeO4Ajn/Ecfxqgs5diat/NIaeoItuGUYepXFoVv8DVL3wpp45E
02hjmp4j2MYNpc6Pc3nPOVntu6YBv+whB0VrsVzeqX89u23FFjTLGbfYrmMQflNi
j8yynGRePIMFhI= root@satellite.example.com
EOF
```

5. Set the date and time you want for the action to take place, then click **Schedule Remote Command**.

Once the key is in place and accessible, all probes that require it should allow **ssh** connections between the monitoring infrastructure and the monitored system. You may then schedule probes requiring the monitoring daemon to run against the newly configured systems.

1.2.3. Configuring the `mysql` package for probes

If your Red Hat Satellite will serve monitoring-entitled client systems against which you wish to run **MySQL** probes, you must configure the **mysql** package on the Red Hat Satellite. See Appendix A, *Probes* for a listing of all available probes.

Subscribe the Satellite to the Red Hat Enterprise Linux Base channel and install the **mysql** package either through the **up2date**, **yum** or Red Hat Network Hosted.

Once finished, your Satellite may be used to schedule MySQL probes.

1.2.4. Enabling Notifications

In addition to viewing probe status within the Red Hat Network interface, you may be notified whenever a probe changes state. This is especially important when monitoring mission-critical production systems. For this reason, Red Hat recommends taking advantage of this feature.

To enable probe notifications within Red Hat Network, you must have identified a mail exchange server and mail domain during installation of your Red Hat Satellite and configured **sendmail** to properly handle incoming mail. See the *Installation* section of the *Red Hat Satellite Installation Guide* for details.

1.2.4.1. Creating Notification Methods

Notifications are sent via a *notification method*, an email or pager address associated with a specific Red Hat Network user. Although the address is tied to a particular user account, it may serve multiple administrators through an alias or mailing list. Each user account can contain multiple notification methods. To create a notification method:

1. Log into the Satellite as either a Satellite Administrator or Monitoring Administrator.

2. Navigate to **Users** and select the username. On the **User Details** page, click on **Notification Methods → create new method**.
3. Enter an intuitive, descriptive label for the method name, such as **DBA day email**, and provide the correct email address. Remember, the labels for all notification methods are available in a single list during probe creation, so they should be unique to your organization.
4. Select the checkbox if you desire abbreviated messages to be sent to the email address. This shorter format contains only the probe state, system hostname, probe name, time of message, and Send ID. The standard, longer format displays additional message headers, system and probe details, and instructions for response.
5. When finished, click **Create Method**. The new method shows up in the **User Details → Notification Methods** tab and the **Notification** page under the top **Monitoring** category. Click its name to edit or delete it.
6. While adding probes, select the **Probe Notifications** checkbox and select the new notification method from the resulting dropdown menu. Notification methods assigned to probes cannot be deleted until they are dis-associated from the probe.

1.2.4.2. Receiving Notifications

If you create notification methods and associate them with probes, you must be prepared to receive them. These notifications come in the form of brief text messages sent to the specified email address. Here is an example of an email notification:

```
Subject: CRITICAL: [hostname]: Satellite: Users at 1
From: "Monitoring Satellite Notification" (rogerthat01@redhat.com)
Date: Mon, 26 Aug 2013 13:42:28 -0800
To: user@organization.com

This is Red Hat Monitoring Satellite notification 01dc8hqw.

Time: Mon Aug 26, 21:42:25 PST
State: CRITICAL
System: [hostname] ([IP address])
Probe: Satellite: Users
Message: Users 6 (above critical threshold of 2)
Notification #116 for Users

Run from: Red Hat Monitoring Satellite
```

As you can see, the longer email notifications contain virtually everything you would need to know about the associated probe. In addition to the probe command, run time, system monitored, and state, the message contains the *Send ID*, which is a unique character string representing the precise message and probe. In the above message, the Send ID is 01dc8hqw.

Note

Since notifications can be generated whenever a probe changes state, simple changes in your network can result in a flood of notifications. Notifications maybe redirected to a specific inbox meant for notifications to avoid issues with priority mail. The next section discusses redirecting notifications.

1.2.4.3. Redirecting Notifications

Upon receiving a notification, you may redirect it by including advanced notification rules within an

acknowledgment email. Enable email reply redirects by opening **/etc/aliases** and adding the following line:

```
rogerthat01:    "| /etc/smrsh/ack_queuer.pl"
```

Once the parameter has been set, reply to the notification email and include the desired option. These are the possible redirect options, or *filter types*:

- ACK METOO - Sends the notification to the redirect destination(s) *in addition to* the default destination.
- ACK SUSPEND - Suspends the notification method for a specified time period.
- ACK AUTOACK - Does not change the destination of the notification, but automatically acknowledges matching alerts as soon as they are sent.
- ACK REDIR - Sends the notification to the redirect destination(s) *instead of* the default destination.

The format of the rule should be ***filter_type probe_type duration email_address*** where ***filter_type*** indicates one of the previous advanced commands, ***probe_type*** indicates **check** or **host**, ***duration*** indicates the length of time for the redirect, and ***email_address*** indicates the intended recipient. For example:

```
ACK METOO host 1h boss@domain.com
```

Capitalization is not required. Duration can be listed in minutes (m), hours (h), or days (d). Email addresses are needed only for redirects (REDIR) and supplemental (METOO) notifications.

The description of the action contained in the resulting email defaults to the command entered by the user. The reason listed is a summary of the action, such as email ack redirect by user@domain.com where user equals the sender of the email.

Note

You can halt or redirect almost all probe notifications by replying to a notification emails with a variation of the command **ack suspend host**. However, you cannot halt Satellite probe notifications by responding to a probe with **ack suspend host** or other redirect responses. These probes require you to change the notifications within the web interface of the Satellite.

1.2.4.4. Deleting Notification Methods

Existing relationships between methods and probes can complicate the process of deleting notification methods. Follow these steps to remove a notification method:

1. Log into the Satellite as a Satellite Administrator or Monitoring Administrator.
2. Navigate to the **Monitoring** → **Notifications** page and click the name of the method to be removed.
3. On the **User** → **User Details** → **Notification Methods** tab, click **delete method**. If the method is not associated with any probes, you are presented with a confirmation page. Click **Confirm Deletion**. The notification method is removed.

Note

Since both the notification method name and address can be edited, consider updating the method rather than deleting it. This redirects notifications from all probes using the method without having to edit each probe and create a new notification method.

4. If the method is associated with one or more probes, you are presented with a list of the probes using the method and the systems to which the probes are attached instead of a confirmation page. Click the probe name to go directly to the **System Details → Probes** tab.
5. Select another notification method and click **Update Probe**.
6. Return to the **Monitoring → Notifications** page and delete the notification method.

1.2.5. About Probes

Now that the Red Hat Network monitoring daemon has been installed and notification methods have been created, you may begin installing probes on your monitoring-entitled systems. If a system is entitled to monitoring, a **Probes** tab appears within its **System Details** page. This is where you will conduct most probe-related work.

1.2.5.1. Managing Probes

Probes are created through the Red Hat Satellite server. Once the probe has been created, the probes are propagated to the specified monitoring-entitled systems registered to the Satellite. Follow the steps below to add a probe in the Satellite server:

1. Log into the Satellite as either a Satellite Administrator or the System Group Administrator for the system.
2. Navigate to the **System Details → Probes** tab and click **create new probe**.
3. On the **System Probe Creation** page, complete all required fields. First, select the Probe Command Group. This alters the list of available probes and other fields and requirements. See Appendix A, *Probes* for the complete list of probes by command group. Remember that some probes require the Red Hat Network monitoring daemon to be installed on the client system.
4. Select the desired Probe Command and the monitoring Scout, typically **Red Hat Monitoring Satellite** but possibly a Red Hat Satellite Proxy Server. Enter a brief but unique description for the probe.
5. Select the **Probe Notifications** checkbox to receive notifications when the probe changes state. Use the **Probe Check Interval** dropdown menu to determine how often notifications should be sent. Selecting **1 minute** (and the **Probe Notification** checkbox) means you will receive notifications every minute the probe surpasses its CRITICAL or WARNING thresholds. See Section 1.2.4, “Enabling Notifications” to find out how to create notification methods and acknowledge their messages.
6. Use the **RHNMD User** and **RHNMD Port** fields, if they appear, to force the probe to communicate via **sshd**, rather than the Red Hat Network monitoring daemon. See Section 1.2.2.2, “Configuring SSH” for details. Otherwise, accept the default values of **nocpulse** and **4545**, respectively.
7. If the **Timeout** field appears, review the default value and adjust to meet your needs. Most but not all timeouts result in an UNKNOWN state. If the probe's metrics are time-based, ensure the timeout is not less than the time allotted to thresholds. Otherwise, the metrics serve no purpose, as the probe will time out before any thresholds are crossed.
8. Use the remaining fields to establish the probe's alert thresholds, if applicable. These CRITICAL and WARNING values determine at what point the probe has changed state. See Section 1.2.5.2,

"Establishing Thresholds" for best practices regarding these thresholds.

9. When finished, click **Create Probe**. Remember, you must commit your monitoring configuration change on the **Scout Config Push** page for this to take effect.

To delete a probe, navigate to its **Current State** page (by clicking the name of the probe from the **System Details** → **Probes** tab), and click **delete probe**. Finally, confirm the deletion.

1.2.5.2. Establishing Thresholds

Many of the probes offered by Red Hat Satellite contain alert thresholds that, when crossed, indicate a change in state for the probe. For instance, the Linux::CPU Usage probe allows you to set CRITICAL and WARNING thresholds for the percent of CPU used. If the monitored system reports 75 percent of its CPU used, and the WARNING threshold is set to 70 percent, the probe will go into a WARNING state. Some probes offer a multitude of such thresholds.

In order to get the most out of your monitoring entitlement and avoid false notifications, Red Hat recommends running your probes without notifications for a time to establish baseline performance for each of your systems. Although the default values provided for probes may suit you, every organization has a different environment that may require altering thresholds.

1.2.5.3. Monitoring the Satellite Server

In addition to monitoring all of your client systems, you may also use Red Hat Network to monitor your Satellite or Proxy. To monitor your Satellite or Proxy, find a system monitored by the server, and go to that system's **System Details** → **Probes** tab.

Click **create new probe** and select the **Satellite** Probe Command Group. Next, complete the remaining fields as you would for any other probe. See Section 1.2.5.1, "Managing Probes" for instructions.

Although the Satellite or Proxy appears to be monitored by the client system, the probe is actually run from the server on itself. Thresholds and notifications work normally.

Note

Any probes that require Red Hat Network monitoring daemon connections cannot be used against a Red Hat Satellite or Red Hat Satellite Proxy Server on which monitoring software is running. This includes most probes in the Linux command group as well as the Log Agent probes and the Remote Program probes. Use the Satellite command group probes to monitor Red Hat Satellites and Red Hat Satellite Proxy Servers. In the case of Proxy scouts, the probes are listed under the system for which they are reporting data.

1.2.6. Monitoring

If you click the **Monitoring** tab on the top navigation bar, the **Monitoring** category and links appear. These pages, which require Monitoring entitlements, enable you to view the results of probes you have set to run against Monitoring-entitled systems and manage the configuration of your monitoring infrastructure.

Initiate the monitoring of a system through the **Probes** tab of the **System Details** page. See Appendix A, *Probes* for the complete list of available probes.

1.2.6.1. Probe Status

Important

The Monitoring entitlement is required to view this tab.

The **Probe Status** page is shown by default when you click **Monitoring** in the top navigation bar.

The **Probe Status** page displays the summary count of probes in the various states and provides a simple interface to find problematic probes quickly. Note that the probe totals in the tabs at the top of the page may not match the numbers of probes displayed in the tables below. The counts at the top include probes for all systems in your organization, while the tables display probes on only those systems to which you have access through the System Group Administrator role. Also, the probe counts displayed here may be out of sync by as much as one minute.

The following list describes each state and identifies the icons associated with them:

- - *Critical* - The probe has crossed a CRITICAL threshold.
- - *Warning* - The probe has crossed a WARNING threshold.
- - *Unknown* - The probe is not able to accurately report metric or state data.
- - *Pending* - The probe has been scheduled but has not yet run or is unable to run.
- - *OK* - The probe is running successfully.

The **Probe Status** page contains tabs for each of the possible states, as well as one that lists all probes. Each table contains columns indicating probe state, the monitored system, the probes used, and the date and time the status was last updated.

In these tables, clicking the name of the system takes you to the **Monitoring** tab of the **System Details** page. Clicking the name of the probe takes you to its **Current State** page. From there, you may edit the probe, delete it, and generate reports based upon its results.

Monitoring data and probe status information that was previously available only through the web interface of the Satellite can now be exported as a CSV file. Click on the **Download CSV** links throughout the Monitoring pages to download CSV files of relevant information. The exported data may include, but is not limited to:

- Probe status
- All probes in a given state (OK, WARN, UNKNOWN, CRITICAL, PENDING)
- A Probe Event history

1.2.6.1.1. Probe Status ⇒ Critical

Important

The Monitoring entitlement is required to view this tab.

The probes that have crossed their CRITICAL thresholds or reached a critical status by some other means. For instance, some probes become critical (rather than unknown) when exceeding their timeout period.

1.2.6.1.2. Probe Status ⇒ Warning

Important

The Monitoring entitlement is required to view this tab.

The probes that have crossed their WARNING thresholds.

1.2.6.1.3. Probe Status ⇒ Unknown

Important

The Monitoring entitlement is required for this feature.

The probes that cannot collect the metrics needed to determine probe state. Most but not all probes enter an unknown state when exceeding their timeout period. This may mean that the timeout period should be increased, or the connection cannot be established to the monitored system.

It is also possible the probes' configuration parameters are not correct and their data cannot be found. Finally, this state may indicate that a software error has occurred.

1.2.6.1.4. Probe Status ⇒ Pending

Important

The Monitoring entitlement is required to view this tab.

The probes whose data have not been received by Red Hat Network. This state is expected for a probe that has just been scheduled but has not yet run. If all probes go into a pending state, your monitoring infrastructure may be failing.

1.2.6.1.5. Probe Status ⇒ OK

Important

The Monitoring entitlement is required to view this tab.

The probes that have run successfully without exception. This is the state desired for all probes.

1.2.6.1.6. Probe Status ⇒ All

Important

The Monitoring entitlement is required to view this tab.

All probes scheduled on systems in your account, listed in alphabetical order by the name of system.

1.2.6.1.7. Current State

Important

The Monitoring entitlement is required to view this tab.

Identifies the selected probe's status and when it last ran, while providing the ability to generate a report on the probe. Although this page is integral to monitoring, it is found under the **Probes** tab within the **System Details** page since its configuration is specific to the system being monitored.

To view a report of the probe's results, choose a relevant duration using the **date** fields and decide whether you would like to see metric data, the state change history or both. To obtain metric data, select the metric(s) on which you wish to see a report, and decide (using the checkboxes) whether the results should be shown in a graph, an event log, or both. Then click **Generate report** at the bottom of the page. If no data exist for the probe's metrics, you are presented with the following message: NO DATA FOR SELECTED TIME PERIOD AND METRIC.

1.2.6.2. Notification

Important

The Monitoring entitlement is required to view this tab.

Identifies the contact methods that have been established for your organization. These methods contain email or pager addresses designated to receive alerts from probes.

The various notification methods available to your organization are listed here on the default **Notification** screen. The methods are listed according to the user to which they apply.

To create a new notification method, click on the name of the user to whom the notification will apply. The user's User Details ⇒ Notification Methods page appears. Click on the title of the notification method to edit the properties of the method.

1.2.6.2.1. Notification ⇒ Filters

Notification filters allow you to create long-term rules that suspend, redirect, or automatically acknowledge standard notifications or send supplemental notifications. This can be helpful in managing verbose or frequent probe communication.

1.2.6.2.1.1. Notification ⇒ Notification Filters ⇒ Active Filters

This is the default screen for the Notification Filters tab. It lists all active filters available for your organization. Click the name of the filter to edit the properties of the filter.

To create a notification filter, click the **create new notification filter** link in the upper right of the screen. Configure each option listed below and click the **Save Filter** button to create the filter.

1. *Description*: Enter a value that allows you to distinguish this filter from others.
2. *Type*: Determine what action the filter should take: redirect, acknowledge, suspend, or supplement the incoming notification.
3. *Send to*: The **Redirect Notification** and **Supplemental Notification** options in step two require an email address to which to send the notifications. The remaining options require no email address.
4. *Scope*: Determine which monitoring components are subject to the filter.

5. *Organization/Scout/Probe*: This option allows you to select the organization, scout(s), or probe(s) to which this filter applies. To select multiple items from the list, hold the **Ctrl** key while clicking the names of the items. To select a range of items, hold the **Shift** key while clicking on the first and last items in the range.
6. *Probes in State*: Select which probe state(s) relate to the filter. For example, you may choose to create a supplemental notification for critical probes only. Uncheck the box to the left of any state you want the filter to ignore.
7. *Notifications sent to*: This is the method to which the notification would be sent if no filter were in place. You may, for example, redirect notifications that would normally go to a user should that individual go on vacation, leaving all other notifications from the probe unchanged.
8. *Match Output*: Select precise notification results by entering a regular expression here. If the "Message:" portion of the notification does not match the regular expression, the filter is not applied.
9. *Recurring*: Select whether a filter runs continuously or on a recurring basis. A recurring filter runs multiple times for a period of time smaller than the duration of the filter. For example, a recurring filter could run for 10 minutes of every hour between the start and end times of the filter. A non-recurring filter runs continuously between the start and end times of the filter.
10. *Beginning*: Enter a date and time for the filter to begin operation.
11. *Ending*: Enter an end date and time for the filter.
12. *Recurring Duration*: How long a recurring filter instance is active. This field, applicable to recurring filters only, begins at the **Beginning** time specified above. Any notification generated outside of the specified duration is not filtered.
13. *Recurring Frequency*: How often the filter activates.

Notification filters cannot be deleted. However, a filter may be canceled by setting the end date to some time in the past. (Note that the end date must be equal to or later than the start date, or the change fails.) Another method is to select a set of filters from the **Active** page and click the **Expire Notification Filters** button in the lower right. These filters are then canceled and appear in the **Expired Filters** tab.

1.2.6.2.1.2. Notification ⇒ Notification Filters ⇒ Expired Filters

This tab lists all notification filters whose end date has passed. Expired filters are stored indefinitely; this allows an organization to recycle useful filters as needed and provides a historical record for troubleshooting.

1.2.6.3. Probe Suites

Probe Suites allow you to configure and apply one or more probes to a system or systems. Probe Suites may be configured once and then applied to any number of systems in a batch. This results in time savings and consistency for Monitoring customers.

To create and apply a Probe Suite, first create an empty Probe Suite, then configure member probes, and finally apply the Suite to selected systems.

1. From the Monitoring ⇒ Probe Suites page, select the **create probe suite** link. Enter an easily distinguishable name for the Probe Suite. You may also choose to add a brief description of the Suite. Click the **Create Probe Suite** button to continue.
2. Add and configure the probes that comprise the Suite. Click the **create new probe** link in the upper right.
3. Configure the probe and click the **Create Probe** button in the lower right. Repeat this process until all desired probes have been added.

Note

Sendmail must be configured correctly on your Red Hat Satellite and each client system to which the Probe Suite is applied must have the **rhnmd** daemon installed and running. See the *Red Hat Satellite Installation Guide* for additional information.

4. On the "Systems" tab, add the systems to which the Probe Suite applies. Click the **add systems to probe suite** link in the upper right of the screen to continue.
5. The next page displays a list of all systems with Monitoring entitlements. Check the box to the left of the system(s) to which you wish to apply the Probe Suite, select the monitoring scout you wish to use, and click the **Add systems to probe suite** button to complete the creation of the Probe Suite.

You can either delete or detach probes from the suite. Detaching a probe disassociates the probes from the suite and converts them to system-specific probes for the specified system. This means that changes to the detached probes only effect that system. Deleting a probe removes it from the Suite for all systems.

To remove probes from the Probe Suite:

1. From the Monitoring ⇒ Probe Suites page, click on the title of the Probe Suite you wish to alter.
2. Select the **Probes** sub-tab.
3. Check the box next to the probe you wish to remove.
4. Click the **Delete probes from Probe Suites** button.

You may also remove a system from the Probe Suite. There are two ways to accomplish this. The first method is to detach the system from the Probe Suite. When you do so, the system still has the same probes assigned to it. However, you now have the ability to configure these probes individually without affecting any other systems.

To detach a system from the suite:

1. From the **Monitoring** ⇒ **Probe Suites** page, click on the title of the Probe Suite you wish to alter.
2. Select the **Systems** sub-tab.
3. Check the box next to the system(s) you wish to remove from the Probe Suite.
4. Click the **Detach System(s) from Probe Suite** button

The second method is to remove the system from the suite. This removes the system from the suite and deletes all running probes from the system.

Note

This action deletes all of the Probe Suites' probes from the system as well as all of the historical Time Series and Event Log data. This action is irreversible.

To remove a system from the Probe Suite and delete all associated probes from the system:

1. From the Monitoring ⇒ Probe Suites page, click on the title of the Probe Suite you wish to alter.
2. Select the **Systems** sub-tab.

3. Check the box next to the system(s) you wish to remove from the Probe Suite.
4. Click the **Remove System(s) from Probe Suite** button.

Finally, as with single Probes, you may download a CSV file containing information about Probe Suites. Click the **Download CSV** link at the bottom of the **Monitoring** ⇒ **Probe Suites** page to download the file.

1.2.6.4. Scout Config Push

Important

The Monitoring entitlement is required to view this tab.

Displays the status of your monitoring infrastructure. Anytime you make a change to your monitoring configuration, such as adding a probe to a system or editing a probe's thresholds, you must reconfigure your monitoring infrastructure. Do this by selecting the Red Hat Network Server's checkbox and clicking **Push Scout Configs**. The table on this page identifies the date and time of requested and completed pushes.

Clicking the name of the server opens its Red Hat Network Monitoring Daemon SSH Public Key. This allows you to copy and paste the SSH key to the systems that are monitored by the scout. This is required in order for the Red Hat Network Monitoring Daemon to connect to the Satellite.

1.2.6.5. General Monitoring Config

Important

The Monitoring entitlement is required to view this tab.

The General Monitoring Config page is in **Admin → Red Hat Satellite Configuration → Monitoring**. It collects information that is universally applicable to your Monitoring infrastructure. Modifying anything on this page causes the Monitoring services on the Red Hat Satellite to reset. It also schedules restart events for the Monitoring services on all Monitoring-enabled Red Hat Satellite Proxy Servers that connect to this Satellite. This is done so that the Monitoring services on these servers immediately reload their configuration.

Typically, the defaults provided in other fields are acceptable, since they are derived from your Satellite installation. Nevertheless, you may use the fields on this page to alter your Monitoring configuration. For instance, you may change your mail exchange server here. This page also allows you to alter the destination of all administrative emails from the Satellite. When finished, click **Update Config**.

1.3. Multiple Satellites

Inter-Satellite Synchronization (ISS) allows a Satellite to synchronize content and permissions from another Satellite instance in a *peer-to-peer* relationship. However, in the following section, a Satellite who receives content will be referred to as a "Slave Satellite" and a Satellite who acts as the source where the content is pulled is called a "Master Satellite". When using ISS to synchronize content, the Slave Satellite instance may have a different setup from that of the Master for non-content entities such as Users and Organizations. The Satellite Administrator on the Slave instance is free to add, remove, and change entities independently from what occurs on the Master instance.

Note

Master and Slave are legacy terms that carry connotations that are *not enforced* by the ISS protocol. Please keep their restricted meanings, as described above, in mind while studying this section.

The ISS feature can be used in different ways depending on the needs of the organization. There are ISS configurations where two Satellites may act as both masters and slaves of each other. This section contains a section on use cases, and how best to set up ISS to suit your organization.

ISS Requirements

The following are the required conditions to be able to use ISS:

- Two or more Red Hat Satellite servers
- At least one Red Hat Satellite populated with at least one channel
- Satellite Administrator privileges on all Satellite systems intended for ISS

1.3.1. Inter-Satellite Synchronization

ISS can be configured manually or by a new tool called **`spacewalk-sync-setup`**. Both methods are effective, and it would be left to the user's choice on which one to use.

1.3.1.1. Manual Configuration

Procedure 1.1. Configuring the Master Satellite Server

With Satellite 5.6, ISS allows the Slave Satellite to duplicate the organizational trust hierarchy and the custom channel permissions from the settings configured on the master. This is accomplished by exporting information about specific organizations from the Master Satellite to the receiving Slave Satellite. The Satellite Administrator on the Slave Satellite can then choose to map the Master Organizations to specific Slave Organizations. Future **`satellite-sync`** operations use this information to assign custom channel ownership to the Slave Organization which is mapped to a specific Master Organization. It can also map the trust relationships between the exposed Master Organization to matching Slave Organizations, creating the equivalent relationships on the Slave.

1. On the Web Interface:
    a. Log in as the Satellite Administrator.
    b. Click **Admin** → **ISS Configuration** → **Master Setup**.
    c. On the top right-hand corner, click **`Add New Slave`**.
    d. Fill in the following information:
        - Slave Fully Qualified Domain Name (FQDN)
        - Allow Slave to Sync? - Choosing this field will allow the Slave Satellite to access this Master Satellite. Otherwise, contact with this Slave will be denied.
        - Sync all orgs to Slave? - Checking this field will synchronize all organizations to the Slave Satellite.

Note

Choosing the **Sync All Orgs to Slave?** option on the Master Setup page will override any specifically selected organizations in the Local Organization table below.

e. Click **Create**.

f. (Optional) Click on any local organization to be exported to the Slave Satellite.

g. Click **Allow Orgs**.

Note

In Satellite 5.5, the Master Satellite used the ***iss_slaves*** parameter in the **/etc/rhn/rhn.conf** file to identify which slaves could contact the Master Satellite. Satellite 5.6 uses the information in the Master Setup page to determine this information.

2. On the Command Line:

   a. Enable the inter-satellite synchronization (ISS) feature in the **/etc/rhn/rhn.conf** file:

   ```
   disable_iss=0
   ```

   b. Save the configuration file, and restart the **httpd** service:

   ```
   service httpd restart
   ```

Procedure 1.2. Configuring Slave Servers

Slave Satellite servers are the machines that will receive content synchronized from the master server.

1. In order to securely transfer content to the slave servers, the **ORG-SSL** certificate from the master server is needed. The certificate can be downloaded over HTTP from the **/pub/** directory of any satellite. The file is called **RHN-ORG-TRUSTED-SSL-CERT**, but can be renamed and placed anywhere in the local filesystem of the slave, such as the **/usr/share/rhn/** directory.
2. Log in to the Slave Satellite as the Satellite Administrator.
3. Click **Admin** → **ISS Configuration** → **Slave Setup**.
4. On the top right-hand corner, click **Add New Master**.
5. Fill in the following information:
   - Master Fully-Qualified Domain Name
   - Default Master?
   - Filename of this Master's CA Certificate - Use the full path of the CA Certificate downloaded in the initial step of this procedure.
6. Click **Add New Master**.

Procedure 1.3. Performing an Inter-Satellite Synchronization

Once the master and slave servers are configured, a synchronization can be performed between them.

- Begin the synchronization by running the **satellite-sync** command:

```
satellite-sync -c your-channel
```

Note

Command line options that are manually provided with the **satellite-sync** command will override any custom settings in the **/etc/rhn/rhn.conf** file.

Procedure 1.4. Mapping the Master Satellite's Exported Organizations to the Slave Satellite's Organizations

Prerequisite

After following the procedures preceding this one, the Master Satellite should show up in the Slave Satellite's Slave Setup under **Admin → ISS Configuration → Slave Setup**. If it does not, please re-check the steps above.

A mapping between organizational names on the master Satellite allows for channel access permissions to be set on the Master Satellite and propagated when content is synced to a Slave Satellite. Not all organization and channel details need to be mapped for all Slave Satellites, Satellite administrators can select which permissions and organizations can be synchronized by allowing or omitting mappings.

To complete the mapping, follow this procedure on the Slave Satellite:

1. Log in as the Satellite Administrator.
2. Click on **Admin → ISS Configuration → Slave Setup**.
3. Select a Master Satellite by clicking on it's name.
4. Use the drop-down box to map the exported master organization name to a matching local organization in the Slave Satellite.
5. Click **Update Mapping**.
6. On the command line, issue the **satellite-sync** on each of the custom channels to obtain the correct trust structure and channel permissions:

```
satellite-sync -c your-channel
```

1.3.1.2. Automated Configuration

spacewalk-sync-setup allows users to specify a Master and Slave Satellite instance and uses configuration files to set up the information described in both the Master and Slave setup. It can create a set of default configuration files if requested. Essentially, it automates the previously setup and mapped configuration for Master-Slave relationships.

Pre-requisites

In order for automated configuration to succeed:

- The *spacewalk-util* package needs to be installed on the system that will issue the command **spacewalk-sync-setup**.
- Existing organizations with custom permissions on the Master Satellite must be present.
- Existing organizations within the Slave Satellite must be present.

Procedure 1.5. Configuring the Master Satellite Server

1. Enable the inter-satellite synchronization (ISS) feature in the **/etc/rhn/rhn.conf** file:

```
disable_iss=0
```

2. Save the configuration file, and restart the **httpd** service:

```
service httpd restart
```

Procedure 1.6. Configuring Slave Servers

Slave Satellite servers are the machines that will have their content synchronized to the master server.

1. In order to securely transfer content to the slave servers, the **ORG-SSL** certificate from the master server is needed. The certificate can be downloaded over HTTP from the **/pub/** directory of any satellite. The file is called **RHN-ORG-TRUSTED-SSL-CERT**, but can be renamed and placed anywhere in the local filesystem of the slave, such as the **/usr/share/rhn/** directory.
2. Log in to the Slave Satellite as the Satellite Administrator.
3. Click **Admin** → **ISS Configuration** → **Slave Setup**.
4. On the top right-hand corner, click **Add New Master**.
5. Fill in the following information:
   - Master Fully-Qualified Domain Name
   - Default Master?
   - Filename of this Master's CA Certificate - Use the full path of the CA Certificate downloaded in the initial step of this procedure.
6. Click **Add New Master**.

Procedure 1.7. Mapping Master Satellite Organizations to Slave Satellite Organizations with spacewalk-sync-setup

1. Log in to a system. It does not matter if it is a Master Satellite, a Slave Satellite or a different system altogether, as long as the system can access the public XMLRPC API of the Master and Slave Satellites.
2. Issue the **spacewalk-sync-setup** on a command line interface:

```
spacewalk-sync-setup --ms=[Master_FQDN] \
--ml=[Master_Sat_Admin_login] \
--mp=[Master_Sat_Admin_password] \
--ss=[Slave FQDN]  --sl=[Slave_Sat_Admin_login] \
--sp=[Slave_Sat_Admin_password> \
--create-templates --apply
```

   Where:
   - --ms=MASTER, --master-server=MASTER is the FQDN of the Master to connect to
   - --ml=MASTER_LOGIN, --master-login=MASTER_LOGIN is the Satellite Administrator login for the Master Satellite
   - --mp=MASTER_PASSWORD, --master-password=MASTER_PASSWORD is the password for the Satellite Administrator login on the Master Satellite
   - --ss=SLAVE, --slave-server=SLAVE is the FQDN of the Slave Satellite to connect to.
   - --sl=SLAVE_LOGIN, --slave-login=SLAVE_LOGIN is the Satellite Administrator login for the

Slave Satellite

- --sp=SLAVE_PASSWORD, --slave-password=SLAVE_PASSWORD is the password for the Satellite Administrator login on the Slave Satellite
- --ct, --create-templates is the option that creates both a master and a slave setup file for the master/slave pair we've pointed at
- --apply tells the Satellite instances to make the changes specified by the setup files to the specified Satellite instances

Note

For more setup options:

```
spacewalk-sync-setup --help
```

The output from this command will be as follows:

```
INFO: Connecting to [admin@master-fqdn]
INFO: Connecting to [admin@slave-fqdn]
INFO: Generating master-setup file $HOME/.spacewalk-sync-setup/master.txt
INFO: Generating slave-setup file $HOME/.spacewalk-sync-setup/slave.txt
INFO: Applying master-setup $HOME/.spacewalk-sync-setup/master.txt
INFO: Applying slave-setup $HOME/.spacewalk-sync-setup/slave.txt
```

3. On the command line, issue the **`satellite-sync`** command on each of the custom channels to obtain the correct trust structure and channel permissions:

```
satellite-sync -c your-channel
```

1.3.2. Organizational Synchronization

Inter-Satellite Synchronization can also be used to import content to any specific organization. This can be done locally or by using remote synchronization. This function is useful for a disconnected satellite with multiple organizations, where content is retrieved through channel dumps or by exporting from connected satellites and then importing it to the disconnected satellite. Organizational synchronization can be used to export custom channels from connected satellites. It can also be used to effectively move content between multiple organizations.

Organizational synchronization follows a clear set of rules in order to maintain the integrity of the source organization:

- If the source content belongs to the ***`NULL`*** organization (that is, it is Red Hat content) it will default to the ***`NULL`*** organization even if a destination organization is specified. This ensures that specified content is always in the privileged ***`NULL`*** organization.
- If an organization is specified at the command line, content will be imported from that organization.
- If no organization is specified, it will default to organization 1.

The following are three example scenarios where organizational IDs (***`orgid`***) are used to synchronize satellites:

Example 1.1. Import Content from Master to Slave Satellite

This example imports content from master to slave satellite:

```
satellite-sync --parent-sat=master.satellite.example.com -c channel-name --
orgid=2
```

Example 1.2. Import Content from an Exported Dump of an Organization

This example imports content from an exported dump of a specific organization:

```
$ satellite-sync -m /dump -c channel-name --orgid=2
```

Example 1.3. Import Content from Red Hat Network Hosted

This example imports content from Red Hat Network Hosted (assuming the system is registered and activated):

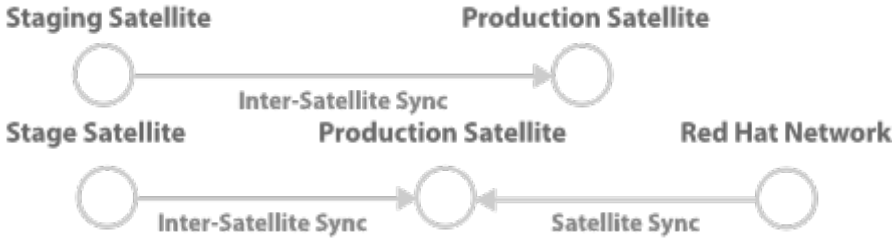
```
$ satellite-sync -c channel-name
```

1.3.3. Inter-Satellite Synchronization Use Cases

Inter-Satellite Synchronization (ISS) can be used in several different ways, depending on the needs of the organization. This section provides examples of how ISS can be used and the methods for setting up and operating these cases.

Example 1.4. Staging Satellite

This example uses one Satellite as a *staging Satellite* to prepare content and perform quality assurance on the packages to ensure they are fit for production use. When content is approved to go to production, the production satellite can synchronize the content from the stage satellite.



1. Run the **satellite-sync** command to synchronize data with **rhn_parent** (usually Red Hat Network Hosted):

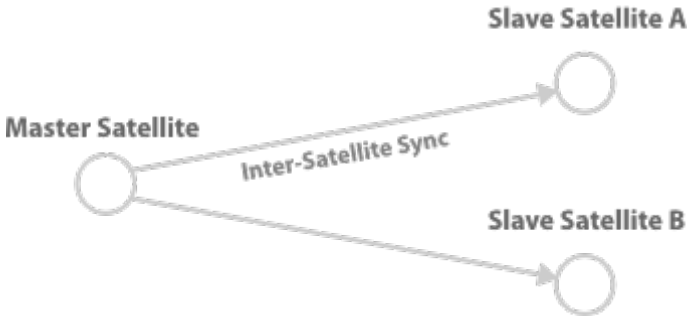
   ```
   satellite-sync -c your-channel
   ```

2. Run the following command to synchronize data from the staging server:

   ```
   satellite-sync --iss-parent=staging-satellite.example.com -c custom-channel
   ```

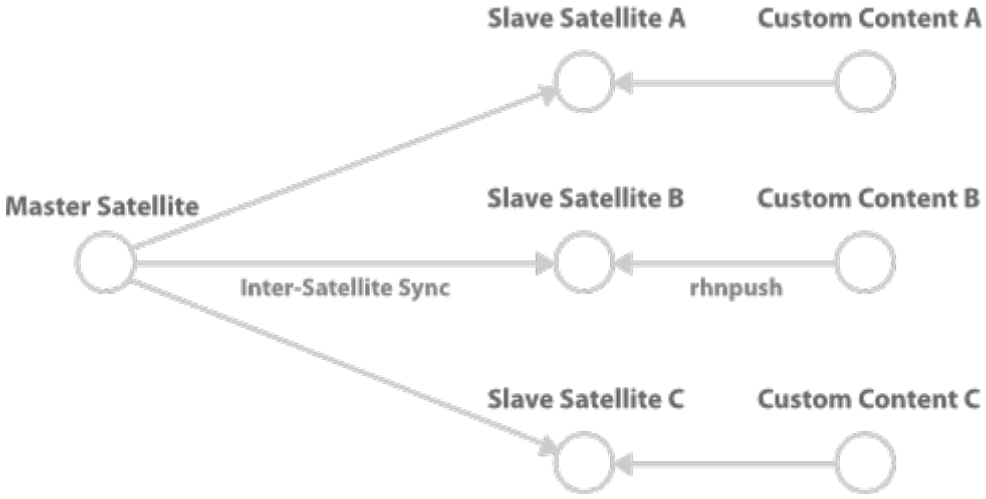
Example 1.5. Synchronized Slaves

In this example, the master satellite provides data directly to the slaves and changes are regularly synchronized.



Example 1.6. Slave Custom Content

This example uses the master satellite as a development channel, from which content is distributed to all production slave satellites. Some of the slave satellites have extra content that is not present in the master satellite channels. These packages are preserved, but all changes from the master satellite are synchronized to the slaves.



Example 1.7. Bi-directional sync

In this environment, two Red Hat Satellite servers act as both master and salve to each other and can synchronize content between them. The Satellite server where the command **`satellite-sync`** is run will pull the content from the other Satellite server and the synchronized data will depend on the options run with **`satellite-sync`**. Without any options, the synchronization will attempt to update everything that was previously synchronized.

See Section 1.3.1.1, “Manual Configuration” for configuring a Master Satellite. Configuring both Satellite servers as a Master will create a bi-directional sync.

Chapter 2. Red Hat Satellite and Solaris-specific Information

This is a section on using Red Hat Satellite with Solaris systems.

2.1. UNIX Support Guide

2.1.1. Introduction

This chapter documents the installation procedure for, and identifies differences in, Red Hat Network functionality when used to manage UNIX-based client systems. Red Hat Network offers UNIX support to help customers migrate from UNIX to Linux. Because of the limited scope of this task, the features offered for UNIX client management are not as comprehensive as those available for managing Red Hat Enterprise Linux systems.

Subsequent sections specify supported UNIX variants, Red Hat Network features supported by the UNIX management system, the prerequisites for managing a UNIX system with Red Hat Network, as well as the installation procedure for UNIX clients.

2.1.1.1. Supported UNIX Variants

The following UNIX variants, versions, and architectures are supported by Red Hat Satellite:

Table 2.1. Supported Solaris Architectures and Versions

Solaris Version	sun4m	sun4d	sun4u	sun4v	sun4us	x86
Solaris 8	yes	no	yes	n/a	no	no
Solaris 9	yes	n/a	yes	n/a	no	yes
Solaris 10	n/a	n/a	yes	yes	no	yes

2.1.1.2. Prerequisites

These items are needed to obtain UNIX support:

- Red Hat Satellite 5.0 or later
- A Satellite certificate with Management entitlements
- Management entitlements for each UNIX client
- Red Hat Network packages for UNIX including python, pyOpenSSL, and the Red Hat Network Client packages
- Sunfreeware packages that provide supporting libraries

Note

Some of these packages are available via the Red Hat Satellite. See Section 2.1.3.1, “Downloading and Installing Additional Packages” for the complete list.

2.1.1.3. Included Features

The following features are included in the UNIX support service level agreement as they exist within Red Hat Network:

- The **Red Hat Network Service Daemon** (**rhnsd**), which triggers **rhn_check** according to a configurable interval

- The **Red Hat Network Configuration Client** (`rhncfg-client`), which executes all configuration actions scheduled from the Satellite
- The **Red Hat Network Configuration Manager** (`rhncfg-manager`), which allows command line administration of Red Hat Network configuration channels
- The `rhn_check` program, which checks in with the Satellite and performs any actions scheduled from the server
- All Management-level functionality, such as system grouping, package profile comparison, and use of the System Set Manager to administer multiple systems at once
- A Provisioning feature called *Remote Command* which enables users to schedule root-level commands on any managed client through the Satellite's website, if the client allows this action

2.1.1.4. Differences in Functionality

The following Red Hat Network features work differently in a UNIX environment:

- The **Red Hat Update Agent for UNIX** offers a much smaller set of options than its Linux counterpart and relies upon the operating system's native toolset for package installation, rather than `rpm` - See Section 2.1.4.2.4, “Updating From the Command Line” for the precise list of options.
- The **Red Hat Network Push** application has been similarly modified to upload native UNIX file types, including packages, patches, and patch clusters.

  Since Solaris package, patch and patch cluster files are different from RPM files, the channel upload mechanism is somewhat different.There are two applications in the `rhnpush` package for Solaris:
  - The first, `solaris2mpm`, is an Red Hat Network utility that creates an MPM file for each Solaris package or patch. The neutral format of the MPM file allows the Satellite to understand and manage the uploaded files.
  - The second, `rhnpush`, has been extended so that it can handle MPM as well as RPM files. Otherwise, it operates identically to the Linux version of `rhnpush`.
- The `Channels` tab of the Red Hat Network website has been augmented to accommodate the storage and installation of native UNIX file types.

2.1.1.5. Excluded Features

The following Red Hat Network features are not available with the UNIX support system:

- All Provisioning-level functionality, such as kickstarting and package rollback, with the exception of configuration file management
- All Errata-related options, since the concept of Errata Updates is not understood in UNIX
- Source files for packages

Answer files are not supported. Support for such files is planned for a future release.

There is also no support for IPv6 for Solaris systems.

Additionally, relocating `RHAT*.pkg` files during installation is not supported.

2.1.2. Satellite Server Preparation/Configuration

Configure the Satellite to support UNIX clients before the required files are available for deployment to the client systems. This can be accomplished in one of two ways, depending on whether you have installed your Satellite server:

1. During the Satellite installation:

   Enable UNIX support on the Satellite by checking the "Enable Solaris Support" box during the

installation process, as pictured:

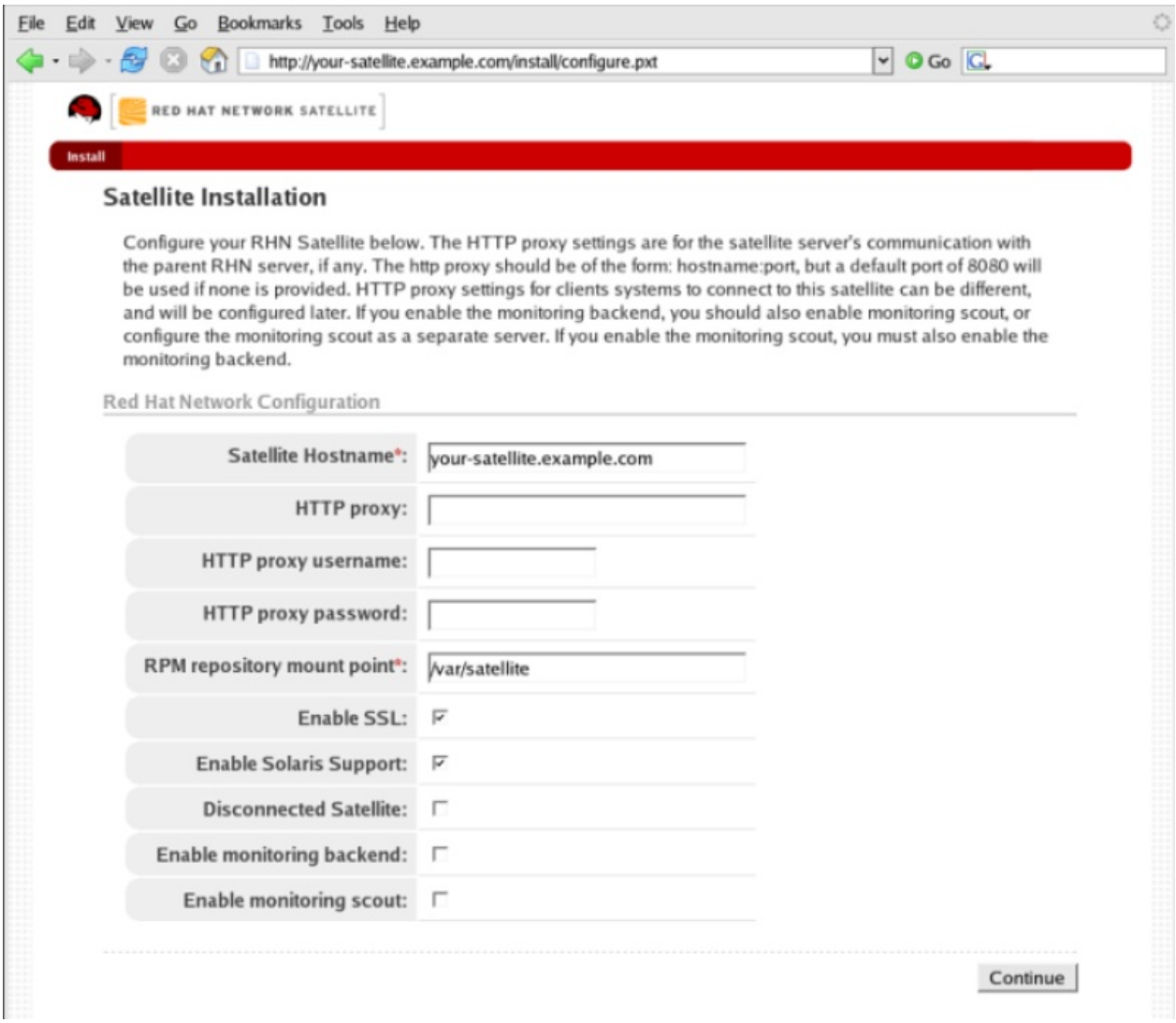

File Edit View Go Bookmarks Tools Help
http://your-satellite.example.com/install/configure.pxt
Go
RED HAT NETWORK SATELLITE
Install
Satellite Installation
Configure your RHN Satellite below. The HTTP proxy settings are for the satellite server's communication with the parent RHN server, if any. The http proxy should be of the form: hostname:port, but a default port of 8080 will be used if none is provided. HTTP proxy settings for clients systems to connect to this satellite can be different, and will be configured later. If you enable the monitoring backend, you should also enable monitoring scout, or configure the monitoring scout as a separate server. If you enable the monitoring scout, you must also enable the monitoring backend.
Red Hat Network Configuration
Satellite Hostname*:
your-satellite.example.com
HTTP proxy:
HTTP proxy username:
HTTP proxy password:
RPM repository mount point*:
/var/satellite
Enable SSL:
Enable Solaris Support:
Disconnected Satellite:
Enable monitoring backend:
Enable monitoring scout:
Continue

Figure 2.1. Enabling UNIX Support During Satellite Installation

2. After the Satellite has been installed:

   Enable UNIX support by configuring the Satellite after it has been installed. To do so, select **Admin** in the top menu bar, then select **Satellite Configuration** in the left navigation bar. In the screen that follows, check the **`Enable Solaris Support`** box, as pictured:

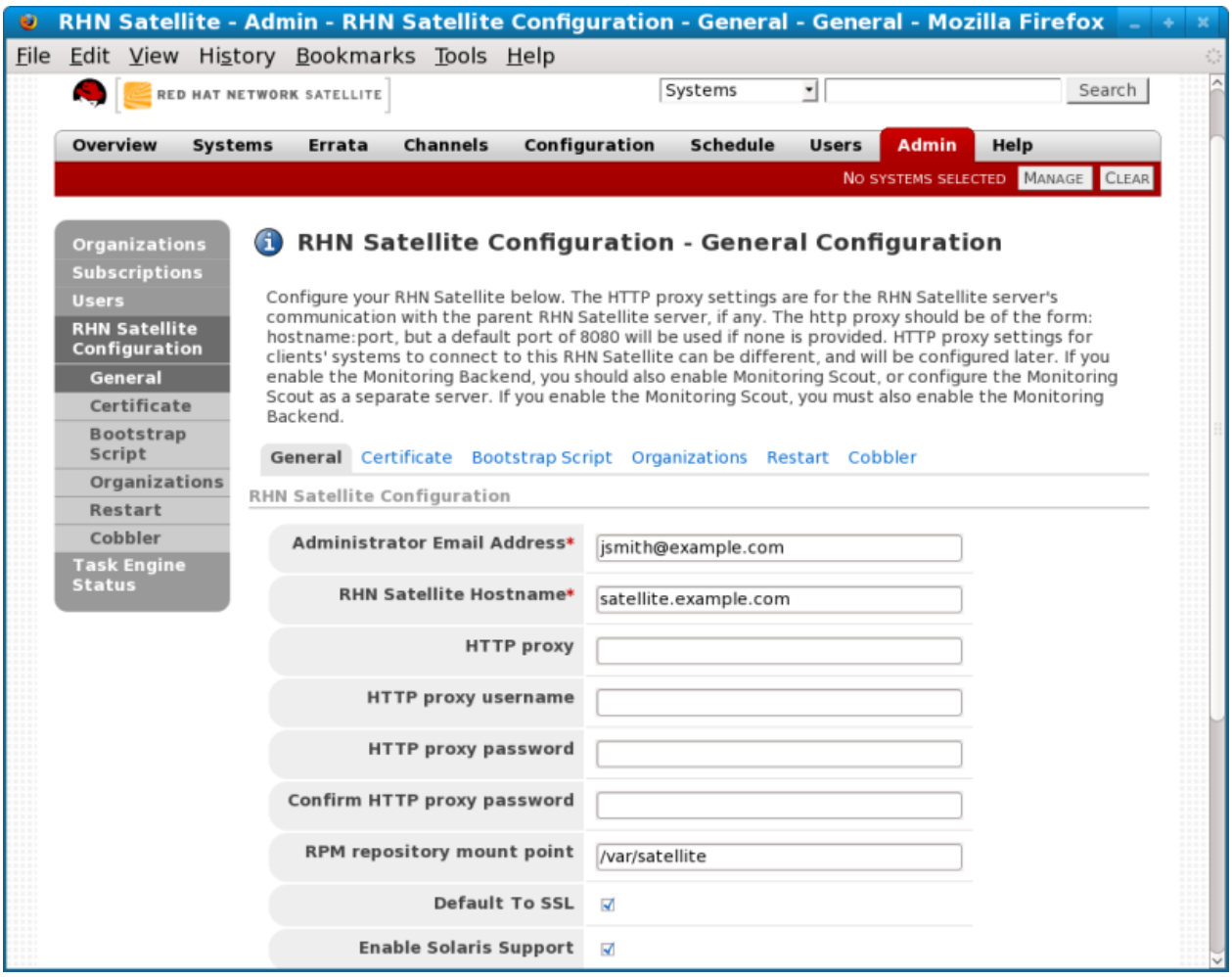

RHN Satellite - Admin - RHN Satellite Configuration - General - General - Mozilla Firefox
File Edit View History Bookmarks Tools Help
RED HAT NETWORK SATELLITE
Systems
Search
Overview Systems Errata Channels Configuration Schedule Users Admin Help
No systems selected Manage Clear
Organizations
Subscriptions
Users
RHN Satellite Configuration
General
Certificate
Bootstrap Script
Organizations
Restart
Cobbler
Task Engine Status
RHN Satellite Configuration - General Configuration
Configure your RHN Satellite below. The HTTP proxy settings are for the RHN Satellite server's communication with the parent RHN Satellite server, if any. The http proxy should be of the form: hostname:port, but a default port of 8080 will be used if none is provided. HTTP proxy settings for clients' systems to connect to this RHN Satellite can be different, and will be configured later. If you enable the Monitoring Backend, you should also enable Monitoring Scout, or configure the Monitoring Scout as a separate server. If you enable the Monitoring Scout, you must also enable the Monitoring Backend.
General Certificate Bootstrap Script Organizations Restart Cobbler
RHN Satellite Configuration
Administrator Email Address*
jsmith@example.com
RHN Satellite Hostname*
satellite.example.com
HTTP proxy
HTTP proxy username
HTTP proxy password
Confirm HTTP proxy password
RPM repository mount point
/var/satellite
Default To SSL
Enable Solaris Support

Figure 2.2. Enabling UNIX Support After Satellite Installation

Click the **Update Configuration** button to confirm the change.

3. Finally, create a base channel to which your client systems may subscribe. Red Hat Network does not provide UNIX content, **satellite-sync** cannot be used to create the channel.

   To create a Solaris channel, login to the web interface of the Satellite as either a Satellite Administrator or a certificate authority. Navigate to the **Channel** tab, followed by the **Manage Software Channels** from the left navigation bar. Click the **create new channel** link in the upper right of the resulting screen. Provide a name and label for your new channel, and select either **SPARC Solaris** or **i386 Solaris** as the architecture, depending on the architecture of the client.

2.1.3. Unix Client System Preparation

Before your UNIX-based client systems benefit from Red Hat Network, they must be prepared for connection:

1. Download and install **gzip** and the required third-party libraries.
2. Download the Red Hat Network application tarball from the Satellite to the client and install the contents.
3. Next, deploy the SSL certificates required for a secure connection.
4. Configure the client applications to connect to the Red Hat Satellite.

Once finished, your systems will be ready to begin receiving Red Hat Network updates. The following

sections explain these steps in detail.

2.1.3.1. Downloading and Installing Additional Packages

This section steps you through the process of downloading and installing third-party applications and the Red Hat Network applications from the Satellite onto the UNIX client.

Of primary importance is the **Red Hat Update Agent for UNIX** (**up2date**), which provides the link between your client systems and Red Hat Network. The UNIX-specific version of the **Red Hat Update Agent** is limited in functionality compared to its Linux counterpart but still enables system registration and facilitates package installs and patches. See Section 2.1.4, “Unix Client Registration and Updates” for a full description of the tool's options.

Note

It may be useful to enter the command **bash** when first logging into the Solaris client. If the BASH shell is available, it will make the system's behavior as Linux-like as possible.

2.1.3.1.1. Install Third-Party Packages

Installation of the Red Hat Network applications cannot proceed unless the following utilities and libraries are present:

- **gzip**
- **libgcc**
- **openssl**
- **zlib**

The **gzip** utility is provided by the SUNW gzip package and may be downloaded from http://www.sunfreeware.com.

On recent versions of Solaris, the necessary libraries are provided by the following natively installed packages:

- **SUNWgccruntime**
- **SUNWopenssl***
- **SUNWzlib**

For older Solaris versions, the following required packages may be downloaded from http://www.sunfreeware.com:

- **SMClibgcc** or **SMCgcc**
- **SMCossl**
- **SMCzlib**

To verify if a package is installed on the client, use the **pkginfo** command. For example, to check for a package that contains "zlib" in the name, run the following command:

```
# pkginfo | grep zlib
```

Note

Solaris package archive names differ from the name of the installed package. For example, the package archive **libgcc<version>-sol<solaris-version>-sparc-local.gz** becomes SMClibgcc after installation

2.1.3.1.2. Configuring the Library Search Path

To allow the Solaris client to use the libraries installed in the previous step, you must add their location to the library search path. To do so, first check the current library search path:

```
# crle -c /var/ld/ld.config
```

Make a note of the current Default Library Path. Next, modify the path to also include the components shown below. Note that the **-l** option resets the value, rather than appending it, so if there already were values set on your system, prepend them to the -l parameter.

On sparc:

```
 # crle -c /var/ld/ld.config -l /other/existing/path:/lib:/usr/lib:/usr/local/lib
```

On x86:

```
# crle -c /var/ld/ld.config -l
/other/existing/path:/lib:/usr/lib:/usr/local/lib:/usr/sfw/lib
```

2.1.3.1.3. Downloading Red Hat Network Client Packages

Download the appropriate tarball of packages from the **/var/www/html/pub/** directory of your Satellite. If you are able to use a GUI web browser like Mozilla, navigate to the **/pub** directory of the Satellite and save the appropriate tarball to your client:

```
http://your-satellite.example.com/pub/rhn-solaris-bootstrap-<version>-<solaris-
arch>-<solaris-version>.tar.gz
```

If you must download the tarball from the command line, it should be possible to use **ftp** to transfer the file from the Satellite to the client.

Using **gzip**, decompress the tarball. You should have the following packages:

- **RHATpossl**
- **RHATrhnrcfg**
- **RHATrhnrcfga**
- **RHATrhnrcfgc**
- **RHATrhnrcfgm**
- **RHATrhnc**
- **RHATrhnl**
- **RHATrpush**
- **RHATsmart**

SMClibgcc and **SMCosslg** may also be included in the tarball.

2.1.3.1.4. Installing the Red Hat Network Packages

Change to the uncompressed directory and use the UNIX variant's native installation tool to install each package. For example, on Solaris, use the **pkgadd** command. Answer "yes" to any prompts during package install.

Here is how a typical installation might proceed:

```
# pkgadd -d RHATpossl-0.6-1.p24.6.pkg all
# pkgadd -d RHATpythn-2.4.1-2.rhn.4.sol9.pkg all
# pkgadd -d RHATrhnl-1.8-7.p23.pkg all
...
```

Note

Use the **-n** option for **pkgadd** to run the command in non-interactive mode. However, this may cause the installation of some packages to fail silently on Solaris 10.

Continue until each package is installed in the Red Hat Network-specific path: **/opt/redhat/rhn/solaris/**.

2.1.3.1.5. Including Red Hat Network Packages in the PATH

In order to make the Red Hat Network packages available at each login, you may wish to add them to your PATH. To do so, add these commands to your login script:

```
# PATH=$PATH:/opt/redhat/rhn/solaris/bin
# PATH=$PATH:/opt/redhat/rhn/solaris/usr/bin
# PATH=$PATH:/opt/redhat/rhn/solaris/usr/sbin
# export PATH
```

To enable access to the Red Hat Network client command man pages, add them to your MANPATH. To do so, add the following commands to your login script:

```
# MANPATH=$MANPATH:/opt/redhat/rhn/solaris/man
# export MANPATH
```

Alternatively, you can also access the man pages from the command line, with the following command:

```
# man -M /opt/redhat/rhn/solaris/man <man page>
```

Finally, add the Red Hat Libraries to your PATH as you did with **libgcc**, **openssl** and **zlib**.

```
crle -c /var/ld/ld.config -l <current library paths>:/opt/redhat/rhn/solaris/lib
```

2.1.3.2. Deploying Client SSL Certificates

To ensure secure data transfer, Red Hat strongly recommends the use of SSL. The Red Hat Satellite eases implementation of SSL by generating the necessary certificates during its installation. The server-side certificate is automatically installed on the Satellite itself, while the client certificate is placed in the **/pub/** directory of the Satellite's Web server.

To install the certificate, follow these steps for each client:

1. Download the SSL certificate from the **/var/www/html/pub/** directory of the Red Hat Satellite onto the client system. The certificate will be named something similar to **RHN-ORG-TRUSTED-SSL-CERT**. It is accessible via the web at the following URL: **https://your-satellite.example.com/pub/RHN-ORG-TRUSTED-SSL-CERT**.
2. Move the client SSL certificate to the Red Hat Network-specific directory for your UNIX variant. For Solaris, this can be accomplished with a command similar to:

```
 mv /path/to/RHN-ORG-TRUSTED-SSL-CERT
/opt/redhat/rhn/solaris/usr/share/rhn/
```

When finished, the new client certificate will be installed in the appropriate directory for your UNIX system. If you have a large number of systems to prepare for Red Hat Network management, you may script this entire process.

Now you must reconfigure the Red Hat Network client applications to refer to the newly installed SSL certificate. See Section 2.1.3.3, “Configuring the clients” for instructions.

2.1.3.3. Configuring the clients

The final step before registering your client systems with Red Hat Network is to reconfigure their Red Hat Network applications to use the new SSL certificate and obtain updates from the Red Hat Satellite. Both of these changes can be made by editing the configuration file of the **Red Hat Update Agent**, which provides registration and update functionality.

Follow these steps on each client system:

1. As root, change to the Red Hat Network configuration directory for the system. For Solaris, the full path is **/opt/redhat/rhn/solaris/etc/sysconfig/rhn/**.
2. Open the **up2date** configuration file in a text editor.
3. Find the ***serverURL*** entry and set its value to the fully qualified domain name (FQDN) of your Red Hat Satellite:

```
serverURL[comment]=Remote server URL
serverURL=https://your-satellite.example.com/XMLRPC
```

4. Ensure the application refers to the Red Hat Satellite even when SSL is turned off by also setting the ***noSSLServerURL*** value to the Satellite:

```
noSSLServerURL[comment]=Remote server URL without SSL
noSSLServerURL=http://your-satellite.example.com/XMLRPC
```

5. With the **up2date** configuration file still open, find the ***sslCACert*** entry and set its value to the name and location of the SSL certificate described in Section 2.1.3.2, “Deploying Client SSL Certificates”, for example:

```
sslCACert[comment]=The CA cert used to verify the ssl server
sslCACert=/opt/redhat/rhn/solaris/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
```

Your client systems are now ready for registration with Red Hat Network and management by your Satellite.

2.1.4. Unix Client Registration and Updates

Now that you have installed Red Hat Network-specific packages, implemented SSL, and reconfigured your client systems to connect to the Red Hat Satellite, you are ready to begin registering systems and obtaining updates.

2.1.4.1. Registering Unix Systems

This section describes the Red Hat Network registration process for UNIX systems. You must use the **rhnreg_ks** command to accomplish this; the use of activation keys for registering your systems is optional. These keys allow you to predetermine settings within Red Hat Network, such as base channels and system groups, and to apply those automatically to systems during their registration.

Since activation key generation and use is covered extensively in other chapters, this section focuses on differences when applying them to UNIX variants.

To register UNIX systems with your Red Hat Satellite, accomplish the following tasks in this order:

1. Log into the Satellite's web interface and click the **Systems** tab in the top navigation bar followed by **Activation Keys** in the left navigation bar. Then click the **create new key** link at the top-right corner of the page.
2. On the following page, select the base channel you created at the end of Section 2.1.2, “Satellite Server Preparation/Configuration”.
3. After creating the key, click its name in the **Activation Keys** list to enhance its Red Hat Network settings by associating software and configuration channels and system groups.
4. Open a terminal on the client system to be registered and switch user to root.
5. Use **rhnreg_ks** along with the **--activationkey** option to register the client with the Satellite. The string of characters that make up the key may be copied directly from the **Activation Keys** list on the website. The resulting command will look something like the following:

   ```
   rhnreg_ks --activationkey=b25fef0966659314ef9156786bd9f3af
   ```

6. Go back to the website, click the name of the activation key, and ensure the new system appears within the **Activated Systems** tab.

2.1.4.2. Obtaining Updates

Package updates in UNIX are handled differently compared to Linux. For instance, Solaris relies on Patch Clusters to update multiple packages at once, while Red Hat operating systems use Errata Updates to associate upgrades with specific packages. In addition, Solaris uses answer files to automate interactive package installations, something Linux doesn't understand, while Red Hat offers the concept of source packages. For this reason, this section seeks to highlight differences in using Red Hat Network tools on UNIX systems. (Note: Red Hat Network does not support Solaris answer files in the current release; such support is planned for future releases.)

Despite inherent differences, such as the lack of Errata, the channel and package management interfaces within the Red Hat Network website on the Satellite work largely the same for UNIX systems. All software channels designed to serve UNIX variants can be constructed almost exactly as the custom channels described in the *Red Hat Satellite Getting Started Guide*. The most significant difference is the architecture. When creating a UNIX software channel, ensure you select the base channel architecture appropriate for the systems to be served.

Break down your packages into base and child channels depending on their nature. For example, on Solaris, installation packages should go in the Solaris base channel, while patches and Patch Clusters should go in a child channel of the Solaris base channel. Extra installation packages can go in a separate Extras child channel.

Red Hat Network treats patches similarly to packages; they are listed and installed in the same way and with the same interface as normal packages. Patches are 'numbered' by Solaris, and will have names like "patch-solaris-108434". The version of a Solaris patch is extracted from the original Solaris metadata, and the release is always 1.

Patch Clusters are bundles of patches that are installed as a unit. Red Hat Network keeps track of the last time that a Patch Cluster was installed successfully on a system. However, Patch Clusters are not tracked on the client as installed entities so they do not appear in the installed packages or patches list. Patch Cluster names look like "patch-cluster-solaris-7_Recommended". The version is a datestring, such as "20040206", the release is always 1 and the epoch is always 0.

2.1.4.2.1. Uploading Packages to the Satellite

Red Hat Network does not provide UNIX content; any Solaris packages, patches or Patch Clusters must be uploaded to the Satellite in a format that it understands from a client system. That package can then be managed and distributed to other systems. Red Hat Network created **`solaris2mpm`** to translate Solaris packages, patches, and patch clusters to a format that the Satellite can understand.

2.1.4.2.1.1. `solaris2mpm`

As mentioned briefly in Section 2.1.1.4, “Differences in Functionality”, **`solaris2mpm`** is part of Red Hat Network Push for Solaris. The content that is pushed to a Solaris channel on the Satellite must first be in .mpm format.

A .mpm file is an archive containing a description of the package data and the package or patch itself. The solaris2mpm command must be run on the client, never the Satellite.

Note

solaris2mpm requires free space equal to three times the size of any package, patch, or patch cluster it is converting. Normally, space in **`/tmp/`** will be used for this purpose. However, the **`--tempdir`** option allows you to specify another directory if necessary.

Multiple files may be specified on the command line of solaris2mpm. Below is a usage example:

```
# solaris2mpm RHATrpush-3.1.5-21.pkg RHATrpush-3.1.5-23.pkg
Opening archive, this may take a while
Writing out RHATrpush-3.1.5-21.sparc-solaris.mpm
Opening archive, this may take a while
Writing out RHATrpush-3.1.5-23.sparc-solaris.mpm
```

Because no other directory was specified, the resulting .mpm files are written to the /tmp/ directory. Note that the name of the resulting .mpm files includes the architecture of the client on which it was created. In this case, this was SPARC Solaris. The general format of mpm file names is:

```
name-version-release.arch.mpm
```

Patch clusters are "exploded" - .mpm files are generated for each patch in the cluster, as well as a top-level "meta" .mpm file containing information about the cluster as a whole.

Below are the options of solaris2mpm:

Table 2.2. solaris2mpm options

Option	Description
--version	Displays the program's version number and exits
-h, --help	Displays this information and exits
-?, --usage	Prints program usage information and exits
--tempdir=<tempdir>	Temporary directory to work from
--select-arch=<arch>	Selects the architecture (i386 or SPARC) for multi-arch packages.

2.1.4.2.1.2. rhnpush with .mpm Files

The Solaris version of **rhnpush** works like the standard utility, but with the added ability to handle .mpm files. Below is a usage example:

```
% rhnpush -v --server testbox.example.com --username myuser -c solaris-8 \
RHATrpush-3.1.5-*.mpm
 Red Hat Network password:
 Connecting to http://testbox.example.com/APP
 Uploading package RHATrpush-3.1.5-21.sparc-solaris.mpm
 Uploading package RHATrpush-3.1.5-23.sparc-solaris.mpm
```

Note

Patch cluster .mpm files must be pushed either concurrently with or after - never before - the .mpm files for the patches contained in that cluster.

Use solaris2mpm on each of the packages, patches, or patch clusters you wish to manage via the Satellite, then use Red Hat Network Push to upload them to the channel created for them.

2.1.4.2.2. Updating Through the Website

To install packages or patches on an individual system, click the name of the system in the **Systems** category, select the packages from the Upgrade or Install lists of the **Packages** or **Patches** tab, and click **Install/Upgrade Selected Packages**.

To run a remote command while installing the package, click **Run Remote Command** rather than **Confirm**. See Section 2.1.5, “Remote Commands” for instructions.

To install packages or patches on multiple systems at once, select the systems and click **System Set Manager** in the left navigation bar. Then, in the **Packages** tab, select the packages from the Upgrade or Install lists and click **Install/Upgrade Packages**. To complete the action, schedule the updates.

2.1.4.2.3. rhnsd

On Red Hat Enterprise Linux systems, the **rhnsd** daemon, which instructs the client system to check in with Red Hat Network, automatically starts at boot time. On Solaris systems, **rhnsd** *does not* start at boot time by default. It can be started from the command line in this way:

```
rhnsd --foreground --interval=240
```

The default location for **rhnsd** is **/opt/redhat/rhn/solaris/usr/sbin/rhnsd**. Below are the

available options for **rhnsd** on Solaris:

Table 2.3. rhnsd Options

Option	Description
-f, --foreground	Run in foreground
-i, --interval=MINS	Connect to Red Hat Network every MINS minutes
-v, --verbose	Log all actions to syslog
-h, --help	Give this help list
-u, --usage	Give this help list
-V, --version	Print program version

2.1.4.2.4. Updating From the Command Line

Like the website, command line use of the **Red Hat Update Agent** is affected by the limitations of UNIX package management. That said, most core functions can still be accomplished through the **up2date** command. The most significant difference is the absence of all options regarding source files. See Table 2.4, "Update Agent Command Line Arguments" for the precise list of options available for UNIX systems.

The command line version of the **Red Hat Update Agent** accepts the following arguments on UNIX systems:

Table 2.4. Update Agent Command Line Arguments

Argument	Description
--version	Show program version information.
-h, **--help**	Show this help message and exit.
-v, **--verbose**	Show additional output.
-l, **--list**	List the latest versions of all packages installed.
-p, **--packages**	Update packages associated with this System Profile.
--hardware	Update this system's hardware profile on Red Hat Network.
--showall	List all packages available for download.
--show-available	List all the packages available that are not currently installed.
--show-orphans	List all the packages currently installed that are not in channels the system is subscribed to.
--show-channels	Show the channel names along with the package names where appropriate.
--installall	Install all available packages. Use with **--channel**.
--channel=*CHANNEL*	Specify which channels to update from using channel labels.
--get	Fetch the package specified without resolving dependencies.

2.1.5. Remote Commands

With UNIX support, Red Hat Network offers the flexibility of issuing remote commands on client systems

through the Satellite's website. This feature allows you to run virtually any (compatible) application or script on any system in your domain without ever having to open a terminal.

2.1.5.1. Enabling Commands

With the flexibility this tool offers comes great risk and the responsibility to mitigate that risk. For all practical purposes, this feature grants a root BASH prompt to anyone with administrative access to the system on the website.

This can be controlled, however, through the same config-enable mechanism used to determine which systems can have their configuration files managed by Red Hat Network.

In short, you must create a directory and file on the UNIX system that tells Red Hat Network it is acceptable to run remote commands on the machine. The directory must be named **`script`**, the file must be named **`run`**, and both must be located in the **`/etc/sysconfig/rhn/allowed-actions/`** directory specific to your UNIX variant.

For instance, in Solaris, issue this command to create the directory:

```
mkdir -p /opt/redhat/rhn/solaris/etc/sysconfig/rhn/allowed-actions/script
```

To create the requisite file in Solaris, issue this command:

```
touch /opt/redhat/rhn/solaris/etc/sysconfig/rhn/allowed-actions/script/run
```

2.1.5.2. Issuing Commands

You may schedule a remote command in a variety of ways: on an individual system, on multiple systems at once, and to accompany a package action.

To run a remote command on an individual system by itself, open the **`System Details`** page and click the **`Remote Command`** subtab. (Note that this subtab only appears if the system has a Provisioning entitlement.) On this page, establish the settings for the command. You may identify a specific user, group, and timeout period, as well as the script itself. Select a date and time to begin attempting the command, and click the **`Schedule Remote Command`** link.

Similarly, you may issue a remote command on multiple systems at once through the **`System Set Manager`**. Select the systems, go to the **`System Set Manager`**, click the **`Provisioning`** tab, and scroll down to the **`Remote Command`** section. From there you may run a remote command on the selected systems at once.

To run a remote command with a package action, schedule the action through the **`Packages`** tab of the **`System Details`** page and click **`Run Remote Command`** while confirming the action. Use the radio buttons at the top to determine whether the command should run before or after the package action, establish the settings for the command, and click **`Schedule Package Install/Upgrade`**.

Note that installing multiple packages that have different remote commands requires scheduling the installs separately or combining the commands into a single script.

Chapter 3. Red Hat Satellite Proxy Information

This is a section on using Red Hat Satellite Proxy with the Red Hat Network Package Manager.

3.1. Using the Red Hat Network Package Manager and Serving Local Packages through the Red Hat Network Proxy

The Red Hat Network Package Manager is a command line tool that allows an organization to serve local packages associated with a private Red Hat Network channel through the Red Hat Network Proxy Server. To update only official Red Hat packages for the Red Hat Network Proxy Server, do not install the Red Hat Network Package Manager.

To use the Red Hat Network Package Manager, install the **`spacewalk-proxy-package-manager`** package and its dependencies.

Only the header information for packages is uploaded to the Red Hat Network Servers. The headers are required so that Red Hat Network can resolve package dependencies for the client systems. The actual package files (**`*.rpm`**) are stored on the Red Hat Network Proxy Server.

The Red Hat Network Package Manager uses the same settings as the Proxy, defined in the **`/etc/rhn/rhn.conf`** configuration file.

Here is a summary of all the command line options for Red Hat Network Package Manager **`rhn_package_manager`**:

Table 3.1. `rhn_package_manager` options

Option	Description
-v, **--verbose**	Increase verbosity.
-d*DIR*, **--dir=*DIR***	Process packages from directory ***DIR***.
-c*CHANNEL*, **--channel=*CHANNEL***	Manage this channel - may be present multiple times.
-n*NUMBER*, **--count=*NUMBER***	Process this number of headers per call - the default is 32.
-l, **--list**	List each package name, version number, release number, and architecture in the specified channel(s).
-s, **--sync**	Check if local directory is in sync with the server.
-p, **--printconf**	Print the current configuration and exit.
-X*PATTERN*, **--exclude=*PATTERN***	Exclude files matching this glob expression - can be present multiple times.
--newest	Push only the packages that are newer than packages already pushed to the server for the specified channel.
--stdin	Read the package names from stdin.
--nosig	Push unsigned packages. By default the Red Hat Network Package Manager attempts to push only signed packages.
--username=*USERNAME*	Specify your Red Hat Network username. If you do not provide one with this option, you will be prompted for it.
--password=*PASSWORD*	Specify your Red Hat Network password. If you do not provide one with this option, you will be prompted for it.
--source	Upload source package headers.
--dontcopy	In the post-upload step, do not copy the packages to their final location in the package tree.
--test	Only print the packages to be pushed.
--no-ssl	*Not recommended* - Turn off SSL.
-?, **--usage**	Briefly describe the options.
--copyonly	Copies the file listed in the argument into the specified channel. Useful when a channel on the proxy is missing a package and you don't want to reimport all of the packages in the channel. E.g., **rhn_package_manager -c*CHANNEL* --copyonly*/PATH/TO/MISSING/FILE***
-h, **--help**	Display the help screen with a list of options.

Tip

These command line options are also described in the **rhn_package_manager** man page: **man rhn_package_manager**.

In order for Red Hat Network Package Manager to be able to serve the local packages, the following steps need to be followed:

1. Create a private channel.
2. Upload the local packages into the channel.

The steps will be further discussed in the next sections.

3.1.1. Creating a Private Channel

Before local packages can be provided through the Red Hat Network Proxy Server, a private channel is needed to store them. Perform the following steps to create a private channel:

1. Log in to the Red Hat Network Web interface at https://rhn.redhat.com or to the local Red Hat Satellite server in the network.
2. Click **Channels** on the top navigation bar. If the **Manage Channels** option is not present in the left navigation bar, ensure that this user has channel editing permissions set. Do this through the **Users** category accessible through the top navigation bar.
3. In the left navigation bar, click **Manage Software Channels** and then the **create new channel** button at the top-right corner of the page.
4. Select a parent channel and base channel architecture, then enter a name, label, summary, and description for the new private channel. The channel label must: be at least six characters long, begin with a letter, and contain only lowercase letters, digits, dashes (-), and periods(.). Also enter the URL of the channel's GPG key. Although this field is not required, it is recommended to enhance security. For instructions on generating GPG keys, see the *Red Hat Network Channel Management Guide*.
5. Click **Create Channel**.

3.1.2. Uploading Packages

Note

You must be an Organization Administrator to upload packages to private Red Hat Network channels. The script will prompt you for your Red Hat Network username and password.

After creating the private channel, upload the package headers for the binary and source RPMs to the Red Hat Network Server and copy the packages to the Red Hat Network Proxy Broker Server. To upload the package headers for the binary RPMs, issue the following command:

```
rhn_package_manager -c "label_of_private_channel" pkg-list
```

This command will upload the header of the package to the channel name specified, and the package itself to **/var/spool/rhn-proxy/rhn**.

pkg-list is the list of packages to be uploaded. Alternatively, use the **-d** option to specify the local directory that contains the packages to add to the channel. Ensure that the directory contains only the packages to be included and no other files. Red Hat Network Package Manager can also read the list of packages from standard input (using **--stdin**).

To upload the package headers for the source RPMs:

```
rhn_package_manager -c "label_of_private_channel" --source pkg-list
```

If you have more than one channel specified (using **-c** or **--channel**), the uploaded package headers will be linked to all the channels listed.

Note

If a channel name is not specified, the packages are not added to any channel. The packages can then be added to a channel using the Red Hat Network web interface. The interface can also be used to modify existing private channels.

After uploading the packages, you can immediately check the Red Hat Network Web interface to verify their presence. Click **Channels** in the top navigation bar, **Manage Software Channels** in the left navigation bar, and then the name of the custom channel. Then click the **Packages** subtab. Each RPM should be listed.

Also check to see if the local directory is in sync with the Red Hat Network Server's image of the channels at the command line:

```
rhn_package_manager -s -c "label_of_private_channel"
```

The **-s** option will list all the missing packages (packages uploaded to the Red Hat Network Server not present in the local directory). You must be an Organization Administrator to use this command. The script will prompt you for your Red Hat Network username and password.

If you are using the Red Hat Network Package Manager to update local packages, you must go to the Red Hat Network website to subscribe the system to the private channel.

Chapter 4. Custom Package Management

This chapter provides an overview of how to build packages for successful delivery via Red Hat Network. Topics covered include why RPM should be used, how to build packages for Red Hat Network, and how to properly sign packages.

4.1. Building Packages for Red Hat Network

Red Hat Network uses the *RPM Package Manager* (RPM) technology to determine what software additions and updates are applicable to each client system. Packages retrieved from Red Hat Network are usually in RPM format. Entire ISO images, however, are available through the **`Software`** tab of the Red Hat Network website, but are not available in Red Hat Satellite installations. If the Satellite server has Solaris support enabled, use Red Hat Network Push to upload Solaris packages to custom channels used by Solaris clients.

RPM is a tool that provides users with a simple method for installing, uninstalling, upgrading, and verifying software packages. It also allows software developers to package the source code and compiled versions of a program for end users and developers.

4.1.1. RPM Benefits

RPM provides the following advantages:

Easy Upgrades

Using RPM, you upgrade individual components of a system without completely reinstalling. When Red Hat releases a new version of Red Hat Enterprise Linux, users do not have to reinstall in order to upgrade. RPM allows intelligent, fully-automated, in-place upgrades of the system. Configuration files in packages are preserved across upgrades so users do not lose customizations. There are no special upgrade files needed to update a package because the same RPM file is used to install and upgrade the package.

Package Querying

RPM provides querying options that allows a search through the entire RPM database for all packages or just for certain files. RPM can also find out what package the file belongs to and where the package came from. The files contained in the package are in a compressed archive, with a custom binary header containing useful information about the package and its contents. RPM queries the headers quickly and easily.

System Verification

Another feature is the ability to verify packages. If there are concerns that a file related to a package was deleted, verify the package to check the status of the files it provides. The verification notifies you of any anomalies. If errors do exist, the files are reinstalled easily. Modified configuration files are preserved during reinstallation.

Pristine Sources

A crucial design goal of RPM is to allow the use of *pristine* software sources, as distributed by the original authors of the software. With RPM, the pristine sources can be packaged, along with any patches that were used, plus complete build instructions. This is an important advantage for several reasons. For instance, if a new version of a program is released, it is unnecessary to start from scratch to make it compile. Looking at the match may allow you to see what you *might* need to do. All the compiled-in defaults and changes made to get the software to build properly are easily visible using this technique.

Keeping sources pristine may seem important only to developers, but it results in higher quality software for end users as well.

4.1.2. Red Hat Network RPM Guidelines

The strength of RPM lies in its ability to define dependencies and identify conflicts accurately. Red Hat Network relies on this aspect of RPM. Red Hat Network offers an automated environment, which means that no manual intervention can take place during the installation of a package. Therefore, when building RPMs for distribution through Red Hat Network, it is imperative to follow these rules:

1. Learn RPM. It is crucial to have a fundamental understanding of the important features of RPM to build packages properly. For more information about RPM, start with the following resources:
   - http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/index.html
   - http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Packagers_Guide/index.html
   - http://www.gurulabs.com/GURULABS-RPM-LAB/GURULABS-RPM-GUIDE-v1.0.PDF
2. When building an RPM for a child channel, build the package on a fresh install of Red Hat Enterprise Linux of the same version as the child's base channel. Be sure to apply all updates from Red Hat Network first.
3. The RPM package must install without using the **`--force`** or **`--nodeps`** options. If an RPM cannot be installed cleanly on a build system, Red Hat Network cannot install it automatically on a system.
4. The RPM package filename must be in the NVR (name, version, release) format and must contain the architecture for the package. The proper format is ***`name-version-release.arch`*`.rpm`**. For example, a valid RPM package filename is **`pkgname-0.84-1.i386.rpm`**, where name is ***`pkgname`***, version is ***`0.84`***, release is ***`1`***, and arch is ***`i386`***.
5. The RPM package should be signed by the maintainer of the package. Unsigned packages may be distributed through Red Hat Network, but the **yum** updater must be forced to accept them. Signing packages is highly recommended and is covered in Section 4.2, “Digital Signatures for Red Hat Network Packages”.
6. If the package is changed in any way, including changing the signature or recompiling, the version or release must be increased incrementally. In other words, the NVRA (including architecture) for each RPM distributed through Red Hat Network must correspond to a unique build to avoid ambiguities.
7. No RPM package may obsolete itself.
8. If a package is split into separate packages, be extremely careful with the dependencies. Do not split an existing package unless there is a compelling reason to do so.
9. No package may rely upon interactive pre-install, post-install, pre-uninstall, or post-uninstall scripts. If the package requires direct user intervention during installation, it cannot work with Red Hat Network.
10. Any pre-install, post-install, pre-uninstall, and post-uninstall scripts should never write anything to stderr or stdout. Redirect the messages to **`/dev/null`** if they are not necessary. Otherwise, write them to a file.
11. When creating the spec file, use the group definitions from **`/usr/share/doc/rpm-<`*`version`*`>/GROUPS`**. If there is not an exact match, select the next best match.
12. Use the RPM dependency feature to make sure the program runs after it is installed.

Important

Do not create an RPM by archiving files and then unarchiving them in the post-install script. This defeats the purpose of RPM.

If the files in the archive are not included in the file list, they cannot be verified or examined for conflicts. In the vast majority of cases, RPM itself can pack and unpack archives most effectively anyway. For instance, don't create files in a %post that cannot or will not be cleaned up in a %postun section.

4.2. Digital Signatures for Red Hat Network Packages

All packages distributed through Red Hat Network should have a *digital signature*. A digital signature is created with a unique private key and can be verified with the corresponding public key. After creating a package, the SRPM (Source RPM) and the RPM can be digitally signed with a GnuPG key. Before the package is installed, the public key is used to verify the package was signed by a trusted party and the package has not changed since it was signed.

4.2.1. Generating a GnuPG Keypair

A GnuPG keypair consists of the private and public keys. To generate a keypair:

1. Type the following command as the root user on the shell prompt:

   ```
   gpg --gen-key
   ```

   GPG Keypairs should not be created by non-root users. The root user can lock memory pages which means the information is never written to disk, unlike non-root users.
2. After executing the command to generate a keypair, an introductory screen containing key options similar to the following will appear:

   ```
   gpg (GnuPG) 2.0.14; Copyright (C) 2009 Free Software Foundation, Inc.
   This is free software: you are free to change and redistribute it.
   There is NO WARRANTY, to the extent permitted by law.

   Please select what kind of key you want:
      (1) RSA and RSA (default)
      (2) DSA and Elgamal
      (3) DSA (sign only)
      (4) RSA (sign only)
   Your selection?
   ```
3. Choose the option: (2) DSA and ElGamal. This option allows you to create a digital signature and encrypt/decrypt with two types of technologies. Type **2** and then press **Enter**.
4. Choose the key size, which is how long the key should be. The longer the key, the more resistant against attacks the messages are. Creating a key of at least 2048 bits in size is recommended.
5. The next option will ask to specify how long the key needs to be valid. When choosing an expiration date, remember that anyone using the public key must also be informed of the expiration and supplied with a new public key. It is recommended to not select an expiration date. If an expiration date is not specified, you are asked to confirm your decision:

   ```
   Key does not expire at all Is this correct (y/n)?
   ```

6. Press **y** to confirm your decision.
7. Provide a User-ID containing your name, your email address, and an optional comment. Each of these is requested individually. When finished, you are presented with a summary of the information you entered.
8. Accept your choices and enter a passphrase.

Note

Like your account passwords, a good passphrase is essential for optimal security in GnuPG. Mix your passphrase with uppercase and lowercase letters, use numbers, and/or include punctuation marks.

9. Once you enter and verify your passphrase, the keys are generated. A message similar to the following appears:

```
We need to generate a lot of random bytes. It is a good idea to perform some
other action (type on the keyboard, move the mouse, utilize the disks)
during the prime generation; this gives the random number generator a
better chance to gain enough entropy.

+++++.+++++.++++++++....+++++++++..+++++.+++++.+++++++.+++++++ +++.
++++++++++++++++++++++++++++++++++++++.........................++++
```

When the activity on the screen ceases, your new keys are placed in the directory **.gnupg** in root's home directory. This is the default location of keys generated by the root user.

To list the root keys, use the command:

```
gpg --list-keys
```

The output is similar to the following:

```
gpg: key D97D1329 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   3  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: next trustdb check due at 2013-08-28
pub   2048D/D97D1329 2013-08-27 [expires: 2013-08-28]
      Key fingerprint = 29C7 2D2A 5F9B 7FF7 6411  A9E7 DE3E 5D0F D97D 1329
uid                  Your Name<you@example.com>
sub   2048g/0BE0820D 2013-08-27 [expires: 2013-08-28]
```

To retrieve the public key, use the following command:

```
gpg --export -a 'Your Name' > public_key.txt
```

The public key is written to the file **public_key.txt**.

This public key is quite important. It's the key that must be deployed to all client systems that receive custom software through **yum**. Techniques for deploying this key across an organization are covered in the *Red Hat Network Client Configuration Guide*.

4.2.2. Signing packages

Before signing packages, configure the **~/.rpmmacros** file to include the following:

```
%_signature gpg
%_gpg_name B7085C8A
```

Replace the **_gpg_name** key ID value of ***B7085C8A*** with the key ID from your GPG keyring that you use to sign packages. This value tells **RPM** which signature to use.

To sign the package ***package-name-1.0-1.noarch.rpm***, use the following command:

```
rpm --resign package-name-1.0-1.noarch.rpm
```

Enter your passphrase. To make sure the package is signed, use the following command:

```
rpm --checksig -v package-name-1.0-1.noarch.rpm
```

Note

Before running the **rpm --checksig -v** command, import the gpg key. See Section 4.3, "Importing Custom GPG Keys" in the next section for more information.

You should see the phrase Good signature from "***Your Name***" in the output, with ***Your Name*** replaced with the name associated with the signing key.

4.3. Importing Custom GPG Keys

For customers who plan to build and distribute their own RPMs securely, it is strongly recommended that all custom RPMs are signed using GNU Privacy Guard (GPG). Generating GPG keys and building GPG-signed packages are covered in theSection 4.2.1, "Generating a GnuPG Keypair".

Once the packages are signed, the public key must be deployed on all systems importing these RPMs. This task has two steps: first, create a central location for the public key so that clients may retrieve it, and second, adding the key to the local GPG keyring for each system.

The first step is common and may be handled using the website approach recommended for deploying Red Hat Network client applications. To do this, create a public directory on the Web server and place the GPG public signature in it:

```
cp /some/path/YOUR-RPM-GPG-KEY /var/www/html/pub/
```

The key can then be downloaded by client systems using **Wget**:

```
wget -O- -q http://your_proxy_or_sat.your_domain.com/pub/YOUR-RPM-GPG-KEY
```

The **-O-** option sends results to standard output while the **-q** option sets **Wget** to run in quiet mode. Remember to replace the ***YOUR-RPM-GPG-KEY*** variable with the filename of your key.

Once the key is available on the client file system, import it into the local GPG keyring. Different operating systems require different methods.

For Red Hat Enterprise Linux 3 or later, use the following command:

```
rpm --import /path/to/YOUR-RPM-GPG-KEY
```

Once the GPG key has been successfully added to the client, the system should be able to validate custom RPMs signed with the corresponding key.

Note

When using custom RPMs and channels, always create a custom GPG key for these packages.
The location of the GPG key also needs to be added to the Kickstart profile.
The custom GPG key needs to be added to the client systems or the Kickstart installation may fail.

Chapter 5. Troubleshooting

This chapter provides tips for determining the cause of and resolving the most common errors associated with Red Hat Satellite. If you need additional help, contact Red Hat Network support at https://access.redhat.com/support/. Log in using your Satellite-entitled account to see the full list of options.

To begin troubleshooting general problems, examine the log file or files related to the component exhibiting failures. A useful exercise is to issue the **`tail -f`** command for all log files and then run **`yum list`**. You should then examine all new log entries for potential clues.

5.1. Disk Space

Q: My disk space filled up fast. What happened and what should I do?

A: A common issue is full disk space. An almost sure sign of this is the appearance of halted writing in the log files. If logging stopped during a write, such as mid-word, the hard disks may be full. To confirm this, run this command and check the percentages in the **`Use%`** column:

```
# df -h
```

In addition to log files, you can obtain valuable information by retrieving the status of your Red Hat Satellite and its various components. This can be done with the command:

```
# /usr/sbin/rhn-satellite status
```

In addition, you can obtain the status of components such as the Apache Web server and the **Red Hat Network Task Engine** individually. For instance, to view the status of the Apache Web server, run the command:

```
# service httpd status
```

5.2. Installing and Updating

Q: SELinux keeps giving me messages when I'm trying to install. Why?

A: If you encounter any issues with SELinux messages (such as AVC denial messages) while installing Red Hat Satellite, be sure to have the **`audit.log`** files available so that Red Hat Support personnel can assist you. You can find the file in **`/var/log/audit/audit.log`** and can attach the file to your Support ticket for engineers to assist you.

Q: I changed `/var/satellite` to an NFS mount, and now SELinux is stopping it from working properly. What do I need to do?

A: SELinux parameters need to be changed based on the new NFS mount in order for SELinux to allow that traffic. Do this with the command:

```
# /usr/sbin/setsebool -P spacewalk_nfs_mountpoint on
```

If you are using Red Hat Enterprise Linux 6, you will also need to run the command:

```
# /usr/sbin/setsebool -P cobbler_use_nfs on
```

Q: My Satellite is failing. Any idea why?

A: Do not subscribe the Red Hat Satellite to any of the following child channels available from Red Hat Network's central servers:

- Red Hat Developer Suite
- Red Hat Application Server
- Red Hat Extras
- JBoss product channels

Subscribing to these channels and updating the Satellite might install newer, incompatible versions of critical software components, causing the Satellite to fail.

5.3. Services

Q: Why isn't the Apache Web server running?

A: If the Apache Web server isn't running, entries in your **/etc/hosts** file may be incorrect.

Q: How do I find out what the status of the Red Hat Network Task Engine is?

A: To obtain the status of the **Red Hat Network Task Engine**, run the command:

```
# service taskomatic status
```

Q: How do I find out what the status of the Satellite's Embedded Database is?

A: To view the status of the Satellite's Embedded Database, if it exists, run the command:

```
# db-control status
```

Q: What do I do if the push capability of the Red Hat Satellite stops working?

A: If the push capability of the Red Hat Satellite ceases to function, it is possible that old log files may be at fault. Stop the jabberd daemon before removing these files. To do so, issue the following commands as root:

```
# service jabberd stop
# rm -f /var/lib/jabberd/db/_db*
# service jabberd start
```

5.4. Connectivity

Q: I can't connect! How do I work out what is wrong?

A: The following measures can be used to troubleshoot general connection errors:

- Attempt to connect to the Red Hat Satellite's database at the command line using the correct connection string as found in **/etc/rhn/rhn.conf**:

```
# sqlplus username/password@sid
```

- Make sure that Red Hat Satellite is using Network Time Protocol (NTP) and set to the appropriate time zone. This also applies to all client systems and the separate database machine in Red Hat Satellite with Stand-Alone Database.
- Confirm the correct package:

  ```
  rhn-org-httpd-ssl-key-pair-MACHINE_NAME-VER-REL.noarch.rpm
  ```

  is installed on the Red Hat Satellite and the corresponding **rhn-org-trusted-ssl-cert-*.noarch.rpm** or raw CA SSL public (client) certificate is installed on all client systems.
- Verify the client systems are configured to use the appropriate certificate.
- If also using one or more Red Hat Satellite Proxy Servers, ensure each Proxy's SSL certificates are prepared correctly. The Proxy should have both its own server SSL key-pair and CA SSL public (client) certificate installed, since it will serve in both capacities. See the SSL Certificates chapter of the *Red Hat Satellite Client Configuration Guide* for specific instructions.
- Make sure client systems are not using firewalls of their own, blocking required ports as identified in the *Red Hat Satellite Installation Guide*'s *Additional Requirements* section.

Q: What do I do if importing or synchronizing a channel fails and I can't recover it?

A: If importing/synchronizing a channel fails and you can't recover it in any other way, run this command to delete the cache:

```
# rm -rf temporary-directory
```

Note

The *Red Hat Satellite Installation Guide* section on *Preparing for Import from Local Media* specifies **/var/rhn-sat-import/** as the temporary directory.

Next, restart the importation or synchronization.

Q: I'm getting "SSL_CONNECT" errors. What do I do now?

A: A common connection problem, indicated by **SSL_CONNECT** errors, is the result of a Satellite being installed on a machine whose time had been improperly set. During the Satellite installation process, SSL certificates are created with inaccurate times. If the Satellite's time is then corrected, the certificate start date and time may be set in the future, making it invalid.

To troubleshoot this, check the date and time on the clients and the Satellite with the following command:

```
# date
```

The results should be nearly identical for all machines and within the "notBefore" and "notAfter" validity windows of the certificates. Check the client certificate dates and times with the following command:

```
# openssl x509 -dates -noout -in /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
```

Check the Satellite server certificate dates and times with the following command:

```
# openssl x509 -dates -noout -in /etc/httpd/conf/ssl.crt/server.crt
```

By default, the server certificate has a one-year life while client certificates are good for 10 years. If you find the certificates are incorrect, you can either wait for the valid start time, if possible, or create new certificates, preferably with all system times set to GMT.

5.5. Logging and Reporting

Q: What are the different log files?

A: Virtually every troubleshooting step should start with a look at the associated log file or files. These provide invaluable information about the activity that has taken place on the device or within the application that can be used to monitor performance and ensure proper configuration. See Table 5.1, “Log Files” for the paths to all relevant log files:

There may be numbered log files (such as /var/log/rhn/rhn_satellite_install.log.1, /var/log/rhn/rhn_satellite_install.log.2, etc.) within the **`/var/log/rhn/`** directory. These are *rotated* logs, which are log files created with a .<NUMBER> extension when the current **`rhn_satellite_install.log`** file fills up to a size as specified by the **`logrotate(8)`** daemon and the contents written to a rotated log file. For example, the **`rhn_satellite_install.log.1`** contains the oldest rotated log file, while **`rhn_satellite_install.log.4`** contains the most recently rotated log.

Table 5.1. Log Files

Component/Task	Log File Location
Apache Web server	**`/var/log/httpd/`** directory
Red Hat Satellite	**`/var/log/rhn/`** directory
Red Hat Satellite Installation Program	**`/var/log/rhn/rhn_satellite_install.log`**
Database installation - *Embedded Database*	**`/var/log/rhn/install_db.log`**
Database population	**`/var/log/rhn/populate_db.log`**
Red Hat Satellite Synchronization Tool	**`/var/log/rhn/rhn_server_satellite.log`**
Monitoring infrastructure	**`/var/log/nocpulse/`** directory
Monitoring notifications	**`/var/log/notification/`** directory
Red Hat Network DB Control - *Embedded Database*	**`/var/log/rhn/rhn_database.log`**
Red Hat Network Task Engine (taskomatic)	**`/var/log/messages`**
yum	**`/var/log/yum.log`**
XML-RPC transactions	**`/var/log/rhn/rhn_server_xmlrpc.log`**

Q: How do I use `spacewalk-report`?

A: There are instances where administrators may need a concise, formatted summary of their Red Hat Satellite resources, whether it is to take inventory of their entitlements, subscribed systems, or

users and organizations. Rather than gathering such information manually from the Satellite interface, Red Hat Satellite includes the **spacewalk-report** command to gather and display vital Satellite information at once.

Note

To use **spacewalk-report** you must have the **spacewalk-reports** package installed.

spacewalk-report allows administrators to organize and display reports about content, errata, systems, system event history, and user resources across the Satellite. The **spacewalk-report** command is used to generate reports on:

- System Inventory - Lists all of the systems registered to the Satellite.
- Entitlements - Lists all organizations on the Satellite, sorted by system or channel entitlements.
- Errata - Lists all the errata relevant to the registered systems, sorts errata by severity as well as the systems that apply to a particular erratum.
- Users - Lists all the users registered to the Satellite, and lists any systems associated with a particular user.
- System History - Lists all, or a subset, of the system events that have occurred.

To get a report in CSV format, run the following at the command prompt of your Satellite server.

```
# spacewalk-report report_name
```

The following reports are available:

Table 5.2. `spacewalk-report` Reports

Report	Invoked as	Description
System Inventory	**`inventory`**	List of systems registered to the server, together with hardware and software information
Entitlements	**`entitlements`**	Lists all organizations on the Satellite with their system or channel entitlements
Errata in channels	**`errata-channels`**	Lists errata in channels
All Errata	**`errata-list-all`**	Complete list of all errata
Errata for systems	**`errata-systems`**	Lists applicable errata and any registered systems that are affected
Users in the system	**`users`**	Lists all users registered to the Satellite
Systems administered	**`users-systems`**	Lists systems that can be administered by individual users
Kickstart Trees	**`kickstartable-trees`**	Lists trees able to be kickstarted
System history	**`system-history`**	Lists system event history
System history channels	**`system-history-channels`**	Lists system event history
System history configuration	**`system-history-configuration`**	Lists system configuration event history
System history entitlements	**`system-history-entitlements`**	Lists system entitlement event history
System history errata	**`system-history-errata`**	Lists system errata event history
System history kickstart	**`system-history-kickstart`**	Lists system kickstart and provisioning event history
System history packages	**`system-history-packages`**	Lists system package event history

For more information about an individual report, run **`spacewalk-report`** with the **`--info`** or **`--list-fields-info`** and the report name. The description and list of possible fields in the report will be shown.

For further information, the **`spacewalk-report(8)`** manpage as well as the **`--help`** parameter of the **`spacewalk-report`** program can be used to get additional information about the program invocations and their options.

.

Q: How do I work out what version of the database schema I have?

A: To determine the version of your database schema, run the command:

```
# rhn-schema-version
```

Q: How do I work out what character set types I have?

A: To derive the character set types of your Satellite's database, run the command:

```
# rhn-charsets
```

Q: Why isn't the administrator getting email?

A: If the administrator is not getting email from the Red Hat Satellite, confirm the correct email addresses have been set for **`traceback_mail`** in **`/etc/rhn/rhn.conf`**.

Q: How do I change the sender of the traceback mail?

A: If the traceback mail is marked from dev-null@rhn.redhat.com and you would like the address to be valid for your organization, include the **`web.default_mail_from`** option and appropriate value in **`/etc/rhn/rhn.conf`**.

5.6. Errors

Q: I'm getting an "Error validating satellite certificate" error during a Red Hat Satellite installation. How do I fix it?

A: An "Error validating satellite certificate" error during a Red Hat Satellite installation is caused by having an HTTP proxy in the environment. This can be confirmed by looking at the **`install.log`** file, and locating the following error:

```
ERROR: unhandled exception occurred:
Traceback (most recent call last):
  File "/usr/bin/rhn-satellite-activate", line 45, in ?
    sys.exit(abs(mod.main() or 0))
  File "/usr/share/rhn/satellite_tools/rhn_satellite_activate.py", line 585,
in main
    activateSatellite_remote(options)
  File "/usr/share/rhn/satellite_tools/rhn_satellite_activate.py", line 291,
in activateSatellite_remote
    ret = s.satellite.deactivate_satellite(systemid, rhn_cert)
  File "/usr/lib/python2.4/site-packages/rhn/rpclib.py", line 603, in
__call__
    return self._send(self._name, args)
  File "/usr/lib/python2.4/site-packages/rhn/rpclib.py", line 326, in _request
    self._handler, request, verbose=self._verbose)
  File "/usr/lib/python2.4/site-packages/rhn/transports.py", line 171, in
request
    headers, fd = req.send_http(host, handler)
  File "/usr/lib/python2.4/site-packages/rhn/transports.py", line 698, in
send_http
    self._connection.connect()
  File "/usr/lib/python2.4/site-packages/rhn/connections.py", line 193, in
connect
    sock.connect((self.host, self.port))
  File "<string>", line 1, in connect
socket.timeout: timed out
```

To resolve the issue:

1. Run the install script in disconnected mode, and skip the database installation which has already been done:

   ```
   # ./install.pl --disconnected --skip-db-install
   ```

2. Open **/etc/rhn/rhn.conf** with your preferred text editor, and add or modify the following line:

   ```
   server.satellite.rhn_parent = satellite.rhn.redhat.com
   ```

   Remove the following line:

   ```
   disconnected=1
   ```

   If you are using a proxy for the connection to Red Hat Network, you will also need to add or modify the following lines to reflect the proxy settings.

   ```
   server.satellite.http_proxy = <hostname>:<port>
   server.satellite.http_proxy_username = <username>
   server.satellite.http_proxy_password = <password>
   ```

3. Re-activate the Satellite in connected mode, using the **rhn-satellite-activate** command as the root user, including the path and filename of the satellite certificate:

   ```
   # rhn-satellite-activate --rhn-cert=/path/to/file.cert
   ```

Alternatively, try running the **install.pl** script in connected mode, but with the **--answer-**

file=*answer file* option. Ensure the answer file has the HTTP proxy information specified as follows:

```
rhn-http-proxy = <hostname>:<port>
rhn-http-proxy-username = <username>
rhn-http-proxy-password = <password>
```

Q: I'm getting an "ERROR: server.mount_point not set in the configuration file" error when I try to activate or synchronize the Red Hat Satellite. How do I fix it?

A: An "ERROR: server.mount_point not set in the configuration file" error during Red Hat Satellite activation or synchronization can occur if the ***mount_point*** configuration parameter in **/etc/rhn/rhn.conf** does not point to a directory path, or the directory path it points to is not present or does not have permission to access the directory.

To resolve the issue, check the value of the ***mount_point*** configuration parameter in **/etc/rhn/rhn.conf**. If it set to the default value of **/var/satellite**, verify that the **/var/satellite** and **/var/satellite/redhat** directories exist. For all values, check that path to the file is accurate, and that the permissions are set correctly.

Q: Why does `cobbler check` give an error saying that it needs a different version of yum-utils?

A: Sometimes, running the **cobbler check** command can give an error similar to the following:

```
# cobbler check
The following potential problems were detected:
#0: yum-utils need to be at least version 1.1.17 for reposync -l, current
version is 1.1.16
```

This is a known issue in Cobbler's **reposync** package. The error is spurious and can be safely ignored. This error will be resolved in future versions of Red Hat Satellite.

Q: I'm getting an "unsupported version" error when I try to activate the Red Hat Satellite certificate. How do I fix it?

A: If your Red Hat Satellite certificate has become corrupted, you could get one of the following errors:

```
ERROR: <Fault -2: 'unhandled internal exception: unsupported version: 96'>
```

```
RHN_PARENT: satellite.rhn.redhat.com
     Error reported from RHN: <Fault -2: 'unhandled internal exception:
unsupported version: 115'>
     ERROR: unhandled XMLRPC fault upon remote activation: <Fault -2:
'unhandled internal exception: unsupported version: 115'>
     ERROR: <Fault -2: 'unhandled internal exception: unsupported version:
115'>
```

```
Invalid satellite certificate
```

To resolve this issue, contact Red Hat support services for a new certificate.

Q: I'm getting an "Internal Server Error" complaining about ASCII when I try to edit the kickstart profile. What's going on?

A: If you have recently added some kernel parameters to your kickstart profile, you might find that when you attempt to **`View a List of Kickstart Profiles`** that you get the following Internal Server Error:

```
'ascii' codec can't encode character u'\u2013'
```

This error occurs because some text in the profile is not being recognized correctly.

To resolve the issue:

1. Ssh directly onto the Satellite server as the root user:

   ```
   # ssh root@satellite.fqdn.com
   ```

2. Find the kickstart profile that is causing the problem by looking at the dates of the files in **`/var/lib/cobbler/config/profiles.d`** and locating the one that was edited most recently:

   ```
   # ls -l /var/lib/cobbler/config/profiles.d/
   ```

3. Open the profile in your preferred text editor, and locate the following text:

   ```
   \u2013hostname
   ```

   Change the entry to read:

   ```
   --hostname
   ```

4. Save changes to the profile and close the file.
5. Restart the Red Hat Satellite services to pick up the updated profile:

```
# rhn-satellite restart
Shutting down rhn-satellite...
Stopping RHN Taskomatic...
Stopped RHN Taskomatic.
Stopping cobbler daemon:                                   [  OK  ]
Stopping rhn-search...
Stopped rhn-search.
Stopping MonitoringScout ...                              [  OK  ]
Stopping Monitoring ...                                   [  OK  ]
Stopping httpd:                                           [  OK  ]
Stopping tomcat5:                                         [  OK  ]
Shutting down osa-dispatcher:                            [  OK  ]
Shutting down Oracle Net Listener ...                    [  OK  ]
Shutting down Oracle DB instance "rhnsat" ...           [  OK  ]
Shutting down Jabber router:                              [  OK  ]
Done.
Starting rhn-satellite...
Starting Jabber services                                  [  OK  ]
Starting Oracle Net Listener ...                         [  OK  ]
Starting Oracle DB instance "rhnsat" ...                [  OK  ]
Starting osa-dispatcher:                                  [  OK  ]
Starting tomcat5:                                         [  OK  ]
Starting httpd:                                           [  OK  ]
Starting Monitoring ...                                   [  OK  ]
Starting MonitoringScout ...                              [  OK  ]
Starting rhn-search...
Starting cobbler daemon:                                  [  OK  ]
Starting RHN Taskomatic...
Done.
```

6. Return to the web interface. Note that the interface can take some time to resolve the services. It should return to normal after some time.

Q: I'm getting "Host Not Found" or "Could Not Determine FQDN" errors. What do I do now?

A: Because Red Hat Network configuration files rely exclusively on fully qualified domain names (FQDNs), it is imperative that key applications are able to resolve the name of the Red Hat Satellite into an IP address. **Red Hat Update Agent**, **Red Hat Network Registration Client**, and the Apache Web server are particularly prone to this problem with the Red Hat Network applications issuing errors of "host not found" and the Web server stating "Could not determine the server's fully qualified domain name" upon failing to start.

This problem typically originates from the **/etc/hosts** file. You may confirm this by examining **/etc/nsswitch.conf**, which defines the methods and the order by which domain names are resolved. Usually, the **/etc/hosts** file is checked first, followed by Network Information Service (NIS) if used, followed by DNS. One of these has to succeed for the Apache Web server to start and the Red Hat Network client applications to work.

To resolve this problem, identify the contents of the **/etc/hosts** file. It may look like this:

```
127.0.0.1 this_machine.example.com this_machine localhost.localdomain \
localhost
```

First, in a text editor, remove the offending machine information, like so:

```
127.0.0.1 localhost.localdomain.com localhost
```

Then, save the file and attempt to re-run the Red Hat Network client applications or the Apache Web server. If they still fail, explicitly identify the IP address of the Satellite in the file, such as:

```
127.0.0.1 localhost.localdomain.com localhost
123.45.67.8 this_machine.example.com this_machine
```

Replace the value here with the actual IP address of the Satellite. This should resolve the problem. Keep in mind, if the specific IP address is stipulated, the file will need to be updated when the machine obtains a new address.

Q: I'm getting a "This server is not an entitled Satellite" when I try to synchronize the Red Hat Satellite server. How do fix it?

A: If **satellite-sync** reports that the server is not activated as a Red Hat Satellite, it isn't subscribed to the respective Red Hat Satellite channel. If this is a newly installed system, make sure that the satellite certificate is activated on the system. If it was activated earlier, then it has become deactivated.

Check the system's child channels to discover if it is subscribed to any Red Hat Network Red Hat Satellite channel. View subscribed channels with the following command:

```
# yum repolist
```

Activate the same Satellite certificate again on your Satellite using this command as the root user:

```
# rhn-satellite-activate -vvv --rhn-cert=/path/to/certificate
```

5.7. Web Interface

Q: I'm having problems with the Red Hat Satellite user interface. Which log files should I check?

A: If you experience errors viewing, scheduling, or working with kickstarts in the Red Hat Satellite user interface, check the **/var/log/tomcat6/catalina.out** log file.

For all other user interface errors, check the **/var/log/httpd/error_log** log file.

5.8. Anaconda

Q: I'm getting an error that says `Error downloading kickstart file`. What is the problem and how do I fix it?

A: This error is usually the result of a network issue. To locate the problem, run the **cobbler check** command, and read the output, which should look something like this:

```
# cobbler check
The following potential problems were detected:
#0: reposync is not installed, need for cobbler reposync, install/upgrade
yum-utils?
#1: yumdownloader is not installed, needed for cobbler repo add with --rpm-
list parameter, install/upgrade yum-utils?
#2: The default password used by the sample templates for newly installed
machines (default_password_crypted in /etc/cobbler/settings) is still set to
'cobbler' and should be changed
#3: fencing tools were not found, and are required to use the (optional) power
management features. install cman to use them
```

If **cobbler check** does not provide any answers, check the following:

- Verify **httpd** is running: **service httpd status**
- Verify **cobblerd** is running: **service cobblerd status**
- Verify that you can fetch the kickstart file using **wget** from a different host:

```
wget http://satellite.example.com/cblr/svc/op/ks/profile/rhel5-i386-
u3:1:Example-Org
```

Q: I'm getting a package installation error that says The file *chkconfig-1.3.30.1-2.i386.rpm* cannot be opened. What is the problem and how do I fix it?

A: Clients will fetch content from Red Hat Satellite based on the ***--url*** parameter in the kickstart. For example:

```
url --url http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-u3
```

If you receive errors from Anaconda stating it can't find images or packages, check that the URL in the kickstart will generate a ***200 OK*** response. You can do this by attempting to **wget** the file located at that URL:

```
wget http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-u3
--2011-08-19 15:06:55--  http://satellite.example.com/ks/dist/ks-rhel-i386-
server-5-u3
Resolving satellite.example.com... 10.10.77.131
Connecting to satellite.example.com|10.10.77.131|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `ks-rhel-i386-server-5-u3.1'
2011-08-19 15:06:55 (0.00 B/s) - `ks-rhel-i386-server-5-u3.1' saved [0/0]
```

If you get a response other than ***200 OK***, check the error logs to find out what the problem is. You can also check the actual file Anaconda tried to download by searching the **access_log** file:

```
# grep chkconfig /var/log/httpd/access_log
10.10.77.131 - - [19/Aug/2011:15:12:36 -0400] "GET
/rhn/common/DownloadFile.do?url=/ks/dist/ks-rhel-i386-server-
5-u3/Server  /chkconfig-1.3.30.1-2.i386.rpm HTTP/1.1" 206 24744 "-"
"urlgrabber/3.1.0 yum/3.2.19"
10.10.76.143 - - [19/Aug/2011:15:12:36 -0400] "GET /ks/dist/ks-rhel-i386-
server-5-u3/Server/chkconfig-
1.3.30.1-2.i386.rpm HTTP/1.1" 206 24744 "-" "urlgrabber/3.1.0 yum/3.2.19"
10.10.76.143 - - [19/Aug/2011:15:14:20 -0400] "GET /ks/dist/ks-rhel-i386-
server-5-u3/Server/chkconfig-
1.3.30.1-2.i386.rpm HTTP/1.1" 200 162580 "-" "urlgrabber/3.1.0 yum/3.2.19"
10.10.77.131 - - [19/Aug/2011:15:14:20 -0400] "GET
/rhn/common/DownloadFile.do?url=/ks/dist/ks-rhel-i386-server-
5-u3/Server/chkconfig-1.3.30.1-2.i386.rpm HTTP/1.1" 200 162580 "-"
"urlgrabber/3.1.0 yum/3.2.19"
```

If the requests are not appearing in the **`access_log`** file, the system might be having trouble with the networking setup. If the requests are appearing but are generating errors, check the error logs.

You can also try manually downloading the files to see if the package is available:

```
wget http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-
u3/Server/chkconfig-1.3.30.1-2.i386.rpm
```

5.9. Tracebacks

Q: I'm getting emails with "WEB TRACEBACK" in the subject. What should I do about them?

A: A typical traceback email might look something like this:

```
Subject: WEB TRACEBACK from satellite.example.com
Date: Wed, 19 Aug 2011 20:28:01 -0400
From:Red Hat Satellite <dev-null@redhat.com>
To: admin@example.com

java.lang.RuntimeException: XmlRpcException calling cobbler.
 at
com.redhat.rhn.manager.kickstart.cobbler.CobblerXMLRPCHelper.invokeMethod(Cobb
lerXMLRPCHelper.java:72)
 at
com.redhat.rhn.taskomatic.task.CobblerSyncTask.execute(CobblerSyncTask.java:76)
 at
com.redhat.rhn.taskomatic.task.SingleThreadedTestableTask.execute(SingleThreaded
TestableTask.java:54)
 at org.quartz.core.JobRunShell.run(JobRunShell.java:203)
 at
org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.java:520)
Caused by: redstone.xmlrpc.XmlRpcException: The response could not be parsed.
 at redstone.xmlrpc.XmlRpcClient.handleResponse(XmlRpcClient.java:434)
 at redstone.xmlrpc.XmlRpcClient.endCall(XmlRpcClient.java:376)
 at redstone.xmlrpc.XmlRpcClient.invoke(XmlRpcClient.java:165)
 at
com.redhat.rhn.manager.kickstart.cobbler.CobblerXMLRPCHelper.invokeMethod(Cobb
lerXMLRPCHelper.java:69)
 ... 4 more
Caused by: java.io.IOException: Server returned HTTP response code: 503 for
URL: http://someserver.example.com:80/cobbler_api
 at
sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.jav
a:1236)
 at redstone.xmlrpc.XmlRpcClient.handleResponse(XmlRpcClient.java:420)
 ... 7 more
```

This indicates that there has been a problem with Cobbler communicating with the **taskomatic** service. Try checking the following:

- Verify **httpd** is running: **# service httpd status**
- Verify **cobblerd** is running: **# service cobblerd status**
- Verify that there are no firewall rules that would prevent **localhost** connections

5.10. Registration

Q: The rhnreg_ks command is failing when I run it, saying ERROR: unable to read system id. What is the problem?

A: At the end of the kickstart file, there is a **%post** section that registers the machine to the Red Hat Satellite:

```
# begin Red Hat management server registration
mkdir -p /usr/share/rhn/
wget http://satellite.example.com/pub/RHN-ORG-TRUSTED-SSL-CERT -O
/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
perl -npe 's/RHNS-CA-CERT/RHN-ORG-TRUSTED-SSL-CERT/g' -i /etc/sysconfig/rhn/*
rhnreg_ks --serverUrl=https://satellite.example.com/XMLRPC --
sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --activationkey=1-
c8d01e2f23c6bbaedd0f6507e9ac079d
# end Red Hat management server registration
```

Interpreting this in the order it was added in, this will:

- Create a directory to house the custom SSL cert used by the Red Hat Satellite.
- Fetch the SSL certificate to use during registration.
- Search and replace the SSL certificate strings from the **rhn-register** configuration files, and then register to the Red Hat Satellite using the SSL certificate and an activation key. Every kickstart profile includes an activation key that assures that the system is assigned the correct base and child channels, and gets the correct system entitlements. If it is a reprovisioning of an existing system, the activation key will also ensure it is associated with the previous system profile.

If the **rhnreg_ks** command fails, you might see errors like this in the **ks-post.log** log file:

```
ERROR: unable to read system id.
```

These errors will also occur if an attempt is made to perform an **rhn_check** and the system has not registered to the Red Hat Satellite.

The best way to troubleshoot this is to view the kickstart file and copy and paste the four steps directly at the command prompt after the kickstart has completed. This will produce error messages that are more detailed to help locate the problem.

5.11. Kickstarts and Snippets

Q: What is the directory structure for kickstarts?

A: The base path where the kickstart files are stored is **/var/lib/rhn/kickstarts/**. Within this directory, raw kickstarts are in the **upload** subdirectory, and wizard-generated kickstarts are in the **wizard** subdirectory:

```
Raw Kickstarts: /var/lib/rhn/kickstarts/upload/$profile_name--$org_id.cfg
Wizard Kickstarts: /var/lib/rhn/kickstarts/wizard/$profile_name--$org_id.cfg
```

Q: What is the directory structure for Cobbler snippets?

A: Cobbler snippets are stored in **/var/lib/rhn/kickstarts/snippets**. Cobbler accesses snippets using the symbolic link **/var/lib/cobbler/snippets/spacewalk**.

```
Snippets:  /var/lib/rhn/kickstarts/snippets/$org_id/$snippet_name
```

Important

Red Hat Satellite RPMs expect the Cobbler kickstart and snippet directories to be in their default locations, do not change them.

5.12. Monitoring

Q: Are there any diagnostic tools that help determine the cause of monitoring errors?

A: Though all monitoring-related activities are conducted through the Satellite interface, Red Hat provides access to some command line diagnostic tools that may help you determine the cause of errors. To use these tools, you must be able to become the **nocpulse** user on the Satellite conducting the monitoring.

First log into the Satellite as root. Then switch to the **nocpulse** user with the following command:

```
su - nocpulse
```

To thoroughly troubleshoot a probe, you must first obtain its probe ID. You may obtain this information by running **rhn-catalog** on the Red Hat Satellite Server as the **nocpulse** user. The output will resemble:

```
2 ServiceProbe on example1.redhat.com (199.168.36.245): test 2
3 ServiceProbe on example2.redhat.com (199.168.36.173): rhel2.1 test
4 ServiceProbe on example3.redhat.com (199.168.36.174): SSH
5 ServiceProbe on example4.redhat.com (199.168.36.175): HTTP
```

The probe ID is the first number, while the probe name (as entered in the Satellite interface) is the final entry on the line. In the above example, the 5 probe ID corresponds to the probe named HTTP.

Further, you may pass the **--commandline** (**-c**) and **--dump** (**-d**) options along with a probe ID to **rhn-catalog** to obtain additional details about the probe, like so:

```
rhn-catalog --commandline --dump 5
```

The **--commandline** option yields the command parameters set for the probe, while **--dump** retrieves everything else, including alert thresholds and notification intervals and methods.

The command above will result in output similar to:

```
5 ServiceProbe on example4.redhat.com (199.168.36.175  ):
linux:cpu usage
      Run as: Unix::CPU.pm --critical=90 --sshhost=199.168.36.175
--warn=70 --timeout=15 --sshuser=nocpulse
--shell=SSHRemoteCommandShell --sshport=4545
```

Now that you have the ID, use it with **rhn-runprobe** to examine the probe's output.

Q: How do I interpret the output of rhn-runprobe?

A: Now that you have obtained the probe ID with **rhn-catalog**, use it in conjunction with **rhn-**

runprobe to examine the complete output of the probe. Note that by default, **rhn-runprobe** works in test mode, meaning no results are entered in the database. Here are its options:

Table 5.3. rhn-runprobe Options

Option	Description
--help	List the available options and exit.
--probe=*PROBE_ID*	Run the probe with this ID.
--prob_arg=*PARAMETER*	Override any probe parameters from the database.
--module=*PERL_MODULE*	Package name of alternate code to run.
--log=all=*LEVEL*	Set log level for a package or package prefix.
--debug=*LEVEL*	Set numeric debugging level.
--live	Execute the probe, queue data and send out notifications (if needed).

At a minimum, include the **--probe** option, the **--log** option, and values for each. The **--probe** option takes the probeID as its value and the **--log** option takes the value "all" (for all run levels) and a numeric verbosity level as its values. Here is an example:

```
rhn-runprobe --probe=5 --log=all=4
```

The above command requests the probe output for probeID 5, for all run levels, with a high level of verbosity.

More specifically, you may provide the command parameters derived from **rhn-catalog**, like so:

```
rhn-runprobe 5 --log=all=4 --sshuser=nocpulse --sshport=4545
```

This yields verbose output depicting the probe's attempted execution. Errors are clearly identified.

5.13. Multi-Organization Satellites and Satellite Certificate

Q: How do I register my systems in a Multiple Organization environment when I do not have enough entitlements in my Satellite Certificate?

A: There are some situations in which you need to free entitlements and do not have a lot of time to do so, and may not have access to each organization in order to do this yourself. There is an option in Multi-Org Satellites that allows the Satellite administrator to reduce an organization's entitlement count below their usage. This method must be done logged into the administrative organization.

For example, logged into the administrative organization, if your certificate is 5 system management entitlements shy of being able to cover all registered systems on your Satellite, the 5 systems that were most recently registered to that organization will be unentitled. This process is described below:

1. In the **/etc/rhn/rhn.conf** file, set ***web.force_unentitlement*** to 1.
2. Restart the Satellite.
3. Reduce the allocated entitlements to the desired organizations either via each organization's **Subscriptions** tab or via individual entitlement's **Organizations** tabs.

4. A number of systems in the organization should now be in an **unentitled** state. The number of systems unentitled in the organization will be equal to the difference between the total number of entitlements you removed from the organization and the number of entitlements the organization did not have applied to the systems.

   For example, if you removed 10 entitlements from the organization in step 3, and the organization has 4 entitlements that were not in use by systems, then 6 systems in the organization will be unentitled.

After you have the sufficient number of entitlements required, you should then be able to activate your new Satellite certificate. Note that modifying the ***web.force_unentitlement*** variable is only necessary to reduce an organization's allocated entitlements below what they are using. If an organization has more entitlements than are being actively used, you do not need to set this variable to remove them.

Q: I have extra entitlements on my Satellite Certificate that are not being used. What happens to these entitlements?

A: If you are issued a new Satellite certificate and it has more entitlements than are being consumed on your Satellite, any extra entitlements will be assigned to the administrative organization. If you log into the web interface as the Satellite administrator, you will be able to allocate these entitlements to other organizations. The previously-allocated entitlements to other organizations will be unaffected.

5.14. Proxy Installation and Configuration

Q: After configuring the Red Hat Network Package Manager how can I determine if the local packages were successfully added to the private Red Hat Network channel?

A: Use the command **`rhn_package_manager -l -c "name_of_private_channel"`** to list the private channel packages known to the Satellite. Or visit the Satellite interface.

After subscribing a registered system to the private channel, you can also execute the command **`yum --disablerepo="*" --enablerepo="your_repo_name" list available`** on the registered system and look for the packages from the private Satellite channel.

Q: How can I determine whether the clients are connecting to the Squid server?

A: The **`/var/log/squid/access.log`** file logs all connections to the Squid server.

Q: The Red Hat Update Agent on the client systems does not connect through the Red Hat Satellite Proxy. How can I resolve this error?

A: Make sure that the latest version of the Red Hat Update Agent is installed on the client systems. The latest version contains features necessary to connect through a Red Hat Satellite Proxy. The latest version can be obtained through the Red Hat Network by issuing the command **`yum update yum`** as root or from http://www.redhat.com/support/errata/.

The Red Hat Satellite Proxy is an extension of Apache. See the *Log Files* section of the *Red Hat Satellite Proxy Installation Guide* for its log file location.

Q: My Red Hat Satellite Proxy configuration does not work. Where do I begin troubleshooting it?

A: Make sure **/etc/sysconfig/rhn/systemid** is owned by root.apache with the permissions 0640.

Read the log files. A list is available on the *Log Files* section of the *Red Hat Satellite Proxy Installation Guide*.

Q: How do I troubleshoot general problems in the Red Hat Satellite Proxy?

A: To begin troubleshooting general problems, examine the log file or files related to the component exhibiting failures.

A common issue is full disk space. An almost sure sign of this is the appearance of halted writing in the log files. If logging stops during a write, such as mid-word, you likely have filled disks. To confirm this, run this command and check the percentages in the Use% column:

```
df -h
```

In addition to log files, you can obtain valuable information by retrieving the status of your various components. This can be done for the Apache Web server and Squid.

To obtain the status of the Apache Web server, run the command:

```
service httpd status
```

To obtain the status of Squid, run the command:

```
service squid status
```

If the administrator is not getting email from the Red Hat Satellite Proxy, confirm the correct email addresses have been set for **traceback_mail** in **/etc/rhn/rhn.conf**.

Q: My Red Hat Satellite Proxy encountered the error "Host Not Found"/"Could not Determine FQDN". What should I do?

A: Because Red Hat Network configuration files rely exclusively on fully qualified domain names (FQDN), it is imperative that key applications are able to resolve the name of the Red Hat Satellite Proxy into an IP address. Red Hat Update Agent, Red Hat Network Registration Client, and the Apache Web server are particularly prone to this problem with the Red Hat Network applications issuing errors of "host not found" and the Web server stating "Could not determine the server's fully qualified domain name" upon failing to start.

This problem originates from the **/etc/hosts** file. Confirm this by examining the **/etc/nsswitch.conf** file, which defines the methods and the order by which domain names are resolved. Usually, the **/etc/hosts** file is checked first, followed by Network Information Service (NIS) if it is being used, followed by DNS. One of these has to succeed for the Apache Web server to start and the Red Hat Network client applications to work.

To resolve this problem, identify the contents of the **/etc/hosts** file. It may look like this:

```
127.0.0.1 this_machine.example.com this_machine localhost.localdomain \
localhost
```

In a text editor, remove the machine host information from the file, it should look like so:

```
127.0.0.1 localhost.localdomain.com localhost
```

Save the file and attempt to re-run the Red Hat Network client applications or the Apache Web server. If they still fail, explicitly identify the IP address of the Proxy in the file, such as:

```
127.0.0.1 localhost.localdomain.com localhost
123.45.67.8 this_machine.example.com this_machine
```

Replace the value here with the actual IP address of the Proxy. This should resolve the problem. Keep in mind, if the specific IP address is stipulated, the file will need to be updated when the machine obtains a new address.

Q: I am having issues with Red Hat Satellite Proxy and network connection errors. What should I do?

A: If you are experiencing problems that you believe to be related to failed connections, follow these measures:

- Confirm the correct package:

  ```
  rhn-org-httpd-ssl-key-pair-MACHINE_NAME-VER-REL.noarch.rpm
  ```

  is installed on the Red Hat Satellite Proxy and the corresponding **rhn-org-trusted-ssl-cert-*.noarch.rpm** or raw CA SSL public (client) certificate is installed on all client systems.
- Verify the client systems are configured to use the appropriate certificate.
- If using one or more Red Hat Satellite Proxies, ensure each Proxy's SSL certificate is prepared correctly. If using the Red Hat Satellite Proxy in conjunction with a Red Hat Satellite, the Proxy should have both its own server SSL key-pair and CA SSL public (client) certificate installed, since it will serve in both capacities. See the SSL Certificates chapter of the *Red Hat Satellite Client Configuration Guide* for specific instructions.
- If the Red Hat Satellite Proxy is connecting through an HTTP Proxy, make sure the URL listed is valid. For instance, the HTTP Proxy URL field should not contain references to protocols, such as http:// or https://. Only the hostname and port should be included in the form hostname:port, such as **your-gateway.example.com:8080**.
- Make sure client systems are not using firewalls of their own, blocking required ports, as identified in the *Additional Requirements* section of the *Red Hat Satellite Proxy Installation Guide*.

Q: I am having issues with package delivery errors and object corruption. What should I check for?

A: If package delivery fails or an object appears to be corrupt, and it is not related to connection errors, you should consider clearing the caches. The Red Hat Satellite Proxy has two caches you should be concerned with: one for Squid and the other for authentication.

The Squid cache is located in **/var/spool/squid/**. To clear it:

1. Stop the Apache Web server: **service httpd stop**
2. Stop the Squid server: **service squid stop**

3. Delete the contents of that directory: **`rm -fv /var/cache/rhn/*`**
4. Restart both services:

```
service squid start
service httpd start
```

The same task can be accomplished quicker by just clearing the directory and restarting squid, but this method will most likely result in a number of Red Hat Network traceback messages.

The internal caching mechanism used for authentication by the Proxy may also need its cache cleared. To do this, issue the following command:

```
rm -fv /var/cache/rhn/*
```

Note

If you have exhausted these troubleshooting steps or want to defer them to Red Hat Network professionals, Red Hat recommends you take advantage of the strong support that comes with Red Hat Satellite. The most efficient way to do this is to aggregate your Satellite's configuration parameters, log files, and database information and send this package directly to Red Hat.
Red Hat Network provides a command line tool explicitly for this purpose: The **Satellite Diagnostic Info Gatherer**, commonly known by its command **satellite-debug**. To use this tool, issue the command as root. You will see the pieces of information collected and the single tarball created, like so:

```
# satellite-debug
Collecting and packaging relevant diagnostic information.
Warning: this may take some time...
    * copying configuration information
    * copying logs
    * querying RPM database (versioning of Red Hat Satellite, etc.)
    * querying schema version and database character sets
    * get diskspace available
    * timestamping
    * creating tarball (may take some time): /tmp/satellite-debug.tar.bz2
    * removing temporary debug tree

Debug dump created, stored in /tmp/satellite-debug.tar.bz2
Deliver the generated tarball to your Red Hat Network contact or support
channel.
```

Once finished, email the new file from the **/tmp/** directory to your Red Hat representative for immediate diagnosis.
Additionally, Red Hat provides a command line tool called the **SoS Report**, commonly known by its command **sosreport**. This tool collects your Proxy's configuration parameters, log files, and database information and sends it directly to Red Hat.
To use this tool for Red Hat Satellite information, you must have the **sos** package installed. Type **sosreport -o rhn** as root on the Satellite server to create a report. For example:

```
[root@satserver ~]# sosreport -o rhn

sosreport (version 1.7)

This utility will collect some detailed  information about the
hardware and  setup of your  Red Hat Enterprise Linux  system.
The information is collected and an archive is  packaged under
/tmp, which you can send to a support representative.
Red Hat will use this information for diagnostic purposes ONLY
and it will be considered confidential information.

This process may take a while to complete.
No changes will be made to your system.

Press ENTER to continue, or CTRL-C to quit.
```

You are then prompted for your first initial and last name, then a support case number.
It may take several minutes for the system to generate and archive the report to a compressed file. Once finished, email the new file from the **/tmp/** directory to your Red Hat representative for immediate diagnosis.

Probes

Monitoring-entitled systems can have probes applied to them that constantly confirm their health and full operability. This section lists the available probes broken down by command group, such as Apache.

Many probes that monitor internal system aspects (such as the Linux::Disk Usage probe) rather than external aspects (such as the Network Services::SSH probe) require the installation of the Red Hat Network monitoring daemon (**rhnmd**). This requirement is noted within the individual probe reference.

Each probe has its own reference in this section that identifies required fields (marked with *), default values, and the thresholds that may be set to trigger alerts. Similarly, the beginning of each command group's section contains information applicable to all probes in that group. Section A.1, “Probe Guidelines” covers general guidelines; the remaining sections examine individual probes.

Note

Nearly all of the probes use *Transmission Control Protocol* (TCP) as their transport protocol. Exceptions to this are noted within the individual probe references.

A.1. Probe Guidelines

The following general guidelines outline the meaning of each probe state, and provide guidance in setting thresholds for your probes.

The following list provides a brief description of the meaning of each probe state:

Unknown

The probes that cannot collect the metrics needed to determine probe state. Most (though not all) probes enter this state when exceeding their timeout period. Probes in this state may be configured incorrectly, as well.

Pending

The probes whose data has not been received by the Red Hat Satellite. It is normal for new probes to be in this state. However, if all probes move into this state, the monitoring infrastructure may be failing.

OK

The probes that have run successfully without error. This is the desired state for all probes.

Warning

The probes that have crossed their WARNING thresholds.

Critical

The probes that have crossed their CRITICAL thresholds or reached a critical status by some other means. (Some probes become critical when exceeding their timeout period.)

While adding probes, select meaningful thresholds that, when crossed, notify you and your administrators of problems within your infrastructure. Timeout periods are entered in seconds unless

otherwise indicated. Exceptions to these rules are noted within the individual probe references.

Important

Some probes have thresholds based on time. In order for such CRITICAL and WARNING thresholds to work as intended, their values cannot exceed the amount of time allotted to the timeout period. Otherwise, an UNKNOWN status is returned in all instances of extended latency, thereby nullifying the thresholds. For this reason, Red Hat strongly recommends ensuring that timeout periods exceed all timed thresholds.

Run your probes without notifications for a time to establish baseline performance for each of your systems. Although the default values provided for probes may suit your needs, every organization has a different environment that may require altering thresholds.

A.2. Apache 1.3.x and 2.0.x

The probes in this section may be applied to instances of the Apache web server. Although the default values presume you will apply these probes using standard HTTP, you may also use them over secure connections by changing the application protocol to **`https`** and the port to **`443`**.

A.2.1. Apache::Processes

The Apache::Processes probe monitors the processes executed on an Apache web server and collects the following metrics:

- Data Transferred Per Child - Records data transfer information about individual children. A child process is one that is created from the parent process or another process.
- Data Transferred Per Slot - The cumulative amount of data transferred by a child process that restarts. The number of slots is configured in the **`httpd.conf`** file using the **`MaxRequestsPerChild`** setting.

The **`ExtendedStatus`** directive in the **`httpd.conf`** file of the Web server must be set to **`On`** for this probe to function properly.

Table A.1. Apache::Processes settings

Field	Value
Application Protocol*	http
Port*	80
Pathname*	/server-status
UserAgent*	NOCpulse-ApacheUptime/1.0
Username	
Password	
Timeout*	15
Critical Maximum Megabytes Transferred Per Child	
Warning Maximum Megabytes Transferred Per Child	
Critical Maximum Megabytes Transferred Per Slot	
Warning Maximum Megabytes Transferred Per Slot	

A.2.2. Apache::Traffic

The Apache::Traffic probe monitors the requests on an Apache web server and collects the following metrics:

- Current Requests - The number of requests being processed by the server at probe runtime.
- Request Rate - The accesses to the server per second since the probe last ran.
- Traffic - The kilobytes per second of traffic the server has processed since the probe last ran.

The **`ExtendedStatus`** directive in the **`httpd.conf`** file of the Web server must be set to **`On`** for this probe to function properly.

Table A.2. Apache::Traffic settings

Field	Value
Application Protocol*	http
Port*	80
Pathname*	/server-status
UserAgent*	NOCpulse-ApacheUptime/1.0
Username	
Password	
Timeout*	15
Critical Maximum Current Requests (number)	
Warning Maximum Current Requests (number)	
Critical Maximum Request Rate (events per second)	
Warning Maximum Request Rate (events per second)	
Critical Maximum Traffic (kilobytes per second)	
Warning Maximum Traffic (kilobytes per second)	

A.2.3. Apache::Uptime

The Apache::Uptime probe stores the cumulative time since the Web server was last started. No metrics are collected by this probe, which is designed to help track service level agreements (SLAs).

Table A.3. Apache::Uptime settings

Field	Value
Application Protocol*	http
Port*	80
Pathname*	/server-status
UserAgent*	NOCpulse-ApacheUptime/1.0
Username	
Password	
Timeout*	15

A.3. BEA WebLogic 6.x and higher

The probes in this section (with the exception of JDBC Connection Pool) can be configured to monitor the properties of any BEA WebLogic 6.x and higher server (Administration or Managed) running on a given host, even in a clustered environment. Monitoring of a cluster is achieved by sending all SNMP queries to the Administration Server of the domain and then querying its Managed Servers for individual data.

In order to obtain this higher level of granularity, the **`BEA Domain Admin Server`** parameter must be used to differentiate between the Administration Server receiving SNMP queries and the Managed Server undergoing the specified probe. If the host to be probed is the Administration Server, then the **`BEA Domain Admin Server`** parameter can be left blank, and both the SNMP queries and the probe will be sent to it only.

If the host to be probed is a Managed Server, then the IP address of the Administration Server should be provided in the **`BEA Domain Admin Server`** parameter, and the Managed Server name should be included in the **`BEA Server Name`** parameter and appended to the end of the **`SNMP Community String`** field. This causes the SNMP queries to be sent to the Administration Server host, as is required, but redirects the specific probe to the Managed Server host.

It should also be noted that the community string needed for probes run against Managed Server hosts should be in the form of **`community_prefix@managed_server_name`** in order for the SNMP query to return results for the desired Managed Server. Finally, SNMP must be enabled on each monitored system. SNMP support can be enabled and configured through the WebLogic Console.

See the documentation that came with your BEA server or information about the BEA website for more details about BEA's community string naming conventions.

A.3.1. BEA WebLogic::Execute Queue

The BEA WebLogic::Execute Queue probe monitors the WebLogic execute queue and provides the following metrics:

- Idle Execute Threads - The number of execution threads in an idle state.
- Queue Length - The number of requests in the queue.
- Request Rate - The number of requests per second.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.4. BEA WebLogic::Execute Queue settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
Queue Name*	default
Critical Maximum Idle Execute Threads	
Warning Maximum Idle Execute Threads	
Critical Maximum Queue Length	
Warning Maximum Queue Length	
Critical Maximum Request Rate	
Warning Maximum Request Rate	

A.3.2. BEA WebLogic::Heap Free

The BEA WebLogic::Heap Free probe collects the following metric:

- Heap Free - The percentage of free heap space.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.5. BEA WebLogic::Heap Free settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
Critical Maximum Heap Free	
Warning Maximum Heap Free	
Warning Minimum Heap Free	
Critical Minimum Heap Free	

A.3.3. BEA WebLogic::JDBC Connection Pool

The BEA WebLogic::JDBC Connection Pool probe monitors the Java Database Connection (JDBC) pool on a domain Admin Server only (no Managed Servers) and collects the following metrics:

- Connections - The number of connections to the JDBC.
- Connections Rate - The speed at which connections are made to the JDBC, measured in connections per second.
- Waiters - The number of sessions waiting to connect to the JDBC.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.6. BEA WebLogic::JDBC Connection Pool settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
JDBC Pool Name*	MyJDBC Connection Pool
Critical Maximum Connections	
Warning Maximum Connections	
Critical Maximum Connection Rate	
Warning Maximum Connection Rate	
Critical Maximum Waiters	
Warning Maximum Waiters	

A.3.4. BEA WebLogic::Server State

The BEA WebLogic::Server State probe monitors the current state of a BEA Weblogic Web server. If the probe is unable to make a connection to the server, a CRITICAL status results.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.7. BEA WebLogic::Server State settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	

A.3.5. BEA WebLogic::Servlet

The BEA WebLogic::Servlet probe monitors the performance of a particular servlet deployed on a WebLogic server and collects the following metrics:

- High Execution Time - The highest amount of time in milliseconds that the servlet takes to execute since the system was started.
- Low Execution Time - The lowest amount of time in milliseconds that the servlet takes to execute since the system was started.
- Execution Time Moving Average - A moving average of the execution time.
- Execution Time Average - A standard average of the execution time.
- Reload Rate - The number of times the specified servlet is reloaded per minute.
- Invocation Rate - The number of times the specified servlet is invoked per minute.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.8. BEA WebLogic::Servlet settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
Servlet Name*	
Critical Maximum High Execution Time	
Warning Maximum High Execution Time	
Critical Maximum Execution Time Moving Average	
Warning Maximum Execution Time Moving Average	

A.4. General

The probes in this section are designed to monitor basic aspects of your systems. When applying them, ensure their timed thresholds do not exceed the amount of time allotted to the timeout period. Otherwise, the probe returns an UNKNOWN status in all instances of extended latency, thereby nullifying the thresholds.

A.4.1. General::Remote Program

The General::Remote Program probe allows you to run any command or script on your system and obtain a status string. Note that the resulting message will be limited to 1024 bytes.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.9. General::Remote Program settings

Field	Value
Command*	
OK Exit Status*	0
Warning Exit Status*	1
Critical Exit Status*	2
Timeout	15

A.4.2. General::Remote Program with Data

The General::Remote Program with Data probe allows you to run any command or script on your system and obtain a value, as well as a status string. To use this probe, you must include XML code in the body of your script. This probe supports the following XML tags:

- <perldata> </perldata>
- <hash> </hash>
- <item key =" "> </item>

The remote program will need to output some iteration of the following code to **STDOUT**:

```
<perldata> <hash> <item
key="data">10</item> <item
key="status_message">status message here</item>
</hash> </perldata>
```

The required value for **data** is the data point to be inserted in the database for time-series trending. The **status_message** is optional and can be whatever text string is desired with a maximum length of 1024 bytes. Remote programs that do not include a **status_message** still report the value and status returned.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe. XML is case-sensitive. The **data** item key name cannot be changed and it must collect a number as its value.

Table A.10. General::Remote Program with Data settings

Field	Value
Command*	
OK Exit Status*	0
Warning Exit Status*	1
Critical Exit Status*	2
Timeout	15

A.4.3. General::SNMP Check

The General::SNMP Check probe tests your SNMP server by specifying a single object identifier (OID) in dotted notation (such as **1.3.6.1.2.1.1.1.0**) and a threshold associated with the return value. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the SNMP server to answer a connection request.

Requirements - SNMP must be running on the monitored system to perform this probe. Only integers can be used for the threshold values.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.11. General::SNMP Check settings

Field	Value
SNMP OID*	
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	2
Timeout*	15
Critical Maximum Value	
Warning Maximum Value	
Warning Minimum Value	
Critical Minimum Value	

A.4.4. General::TCP Check

The General::TCP Check probe tests your TCP server by verifying that it can connect to a system via the specified port number. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the TCP server to answer a connection request.

The probe passes the string specified in the **Send** field upon making a connection. The probe anticipates a response from the system, which should include the substring specified in the **Expect** field. If the expected string is not found, the probe returns a CRITICAL status.

Table A.12. General::TCP Check settings

Field	Value
Send	
Expect	
Port*	1
Timeout*	10
Critical Maximum Latency	
Warning Maximum Latency	

A.4.5. General::UDP Check

The General::UDP Check probe tests your UDP server by verifying that it can connect to a system via the specified port number. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the UDP server to answer a connection request.

The probe passes the string specified in the **Send** field upon making a connection. The probe anticipates a response from the system, which should include the substring specified in the **Expect** field. If the expected string is not found, the probe returns a CRITICAL status.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.13. General::UDP Check settings

Field	Value
Port*	1
Send	
Expect	
Timeout*	10
Critical Maximum Latency	
Warning Maximum Latency	

A.4.6. General::Uptime (SNMP)

The General::Uptime (SNMP) probe records the time since the device was last started. It uses the SNMP object identifier (OID) to obtain this value. The only error status it will return is UNKNOWN.

Requirements - SNMP must be running on the monitored system and access to the OID must be enabled to perform this probe.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.14. General::Uptime (SNMP) settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	2
Timeout*	15

A.5. Linux

The probes in this section monitor essential aspects of your Linux systems, from CPU usage to virtual memory. Apply them to mission-critical systems to obtain warnings prior to failure.

Unlike other probe groups, which may or may not require the Red Hat Network monitoring daemon, every Linux probe requires that the **`rhnmd`** daemon be running on the monitored system.

A.5.1. Linux::CPU Usage

The Linux::CPU Usage probe monitors the CPU utilization on a system and collects the following metric:

- CPU Percent Used - The five-second average of the percent of CPU usage at probe execution.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to run this probe.

Table A.15. Linux::CPU Usage settings

Field	Value
Timeout*	15
Critical Maximum CPU Percent Used	
Warning Maximum CPU Percent Used	

A.5.2. Linux::Disk IO Throughput

The Linux::Disk IO Throughput probe monitors a given disk and collects the following metric:

- Read Rate - The amount of data that is read in kilobytes per second.
- Write Rate - The amount of data that is written in kilobytes per second.

To obtain the value for the required **`Disk number or disk name`** field, run **`iostat`** on the system to be monitored and see what name has been assigned to the disk you desire. The default value of **`0`** usually provides statistics from the first hard drive connected directly to the system.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe. Also, the **`Disk number or disk name`** parameter must match the format visible when the **`iostat`** command is run. If the format is not identical, the configured probe enters an UNKNOWN state.

Table A.16. Linux::Disk IO Throughput settings

Field	Value
Disk number or disk name*	0
Timeout*	15
Critical Maximum KB read/second	
Warning Maximum KB read/second	
Warning Minimum KB read/second	
Critical Minimum KB read/second	
Critical Maximum KB written/second	
Warning Maximum KB written/second	
Warning Minimum KB written/second	
Critical Minimum KB written/second	

A.5.3. Linux::Disk Usage

The Linux::Disk Usage probe monitors the disk space on a specific file system and collects the following metrics:

- File System Used - The percentage of the file system currently in use.
- Space Used - The amount of the file system in megabytes currently in use.
- Space Available - The amount of the file system in megabytes currently available.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.17. Linux::Disk Usage settings

Field	Value
File system*	/dev/hda1
Timeout*	15
Critical Maximum File System Percent Used	
Warning Maximum File System Percent Used	
Critical Maximum Space Used	
Warning Maximum Space Used	
Warning Minimum Space Available	
Critical Minimum Space Available	

A.5.4. Linux::Inodes

The Linux::Inodes probe monitors the specified file system and collects the following metric:

- Inodes - The percentage of inodes currently in use.

An inode is a data structure that holds information about files in a Linux file system. There is an inode for each file, and a file is uniquely identified by the file system on which it resides and its inode number on that system.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.18. Linux::Inodes settings

Field	Value
File system*	/
Timeout*	15
Critical Maximum Inodes Percent Used	
Warning Maximum Inodes Percent Used	

A.5.5. Linux::Interface Traffic

The Linux::Interface Traffic probe measures the amount of traffic into and out of the specified interface (such as eth0) and collects the following metrics:

- Input Rate - The traffic in bytes per second going into the specified interface.
- Output Rate - The traffic in bytes per second going out of the specified interface.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.19. Linux::Interface Traffic settings

Field	Value
Interface*	
Timeout*	30
Critical Maximum Input Rate	
Warning Maximum Input Rate	
Warning Minimum Input Rate	
Critical Minimum Input Rate	
Critical Maximum Output Rate	
Warning Maximum Output Rate	
Warning Minimum Output Rate	
Critical Minimum Output Rate	

A.5.6. Linux::Load

The Linux::Load probe monitors the CPU of a system and collects the following metric:

- Load - The average load on the system CPU over various periods.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.20. Linux::Load settings

Field	Value
Timeout*	15
Critical CPU Load 1-minute average	
Warning CPU Load 1-minute average	
Critical CPU Load 5-minute average	
Warning CPU Load 5-minute average	
Critical CPU Load 15-minute average	
Warning CPU Load 15-minute average	

A.5.7. Linux::Memory Usage

The Linux::Memory Usage probe monitors the memory on a system and collects the following metric:

- RAM Free - The amount of free random access memory (RAM) in megabytes on a system.

You can also include the reclaimable memory in this metric by entering **`yes`** or **`no`** in the **`Include reclaimable memory`** field.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.21. Linux::Memory Usage settings

Field	Value
Include reclaimable memory	no
Timeout*	15
Warning Maximum RAM Free	
Critical Maximum RAM Free	

A.5.8. Linux::Process Counts by State

The Linux::Process Counts by State probe identifies the number of processes in the following states:

- Blocked - A process that has been switched to the waiting queue and whose state has been switched to **`waiting`**.
- Defunct - A process that has terminated (either because it has been killed by a signal or because it has called **`exit()`**) and whose parent process has not yet received notification of its termination by executing some form of the **`wait()`** system call.
- Stopped - A process that has been stopped before its execution could be completed.
- Sleeping - A process that is in the **`Interruptible`** sleep state and that can later be reintroduced into memory, resuming execution where it left off.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.22. Linux::Process Counts by State settings

Field	Value
Timeout*	15
Critical Maximum Blocked Processes	
Warning Maximum Blocked Processes	
Critical Maximum Defunct Processes	
Warning Maximum Defunct Processes	
Critical Maximum Stopped Processes	
Warning Maximum Stopped Processes	
Critical Maximum Sleeping Processes	
Warning Maximum Sleeping Processes	
Critical Maximum Child Processes	
Warning Maximum Child Processes	

A.5.9. Linux::Process Count Total

The Linux::Process Count Total probe monitors a system and collects the following metric:

- Process Count - The total number of processes currently running on the system.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.23. Linux::Process Count Total settings

Field	Value
Timeout*	15
Critical Maximum Process Count	
Warning Maximum Process Count	

A.5.10. Linux::Process Health

The Linux::Process Health probe monitors user-specified processes and collects the following metrics:

- CPU Usage - The CPU usage rate for a given process in milliseconds per second. This metric reports the time column of **ps** output, which is the cumulative CPU time used by the process. This makes the metric independent of probe interval, allows sane thresholds to be set, and generates usable graphs (i.e. a sudden spike in CPU usage shows up as a spike in the graph).
- Child Process Groups - The number of child processes spawned from the specified parent process. A child process inherits most of its attributes, such as open files, from its parent.
- Threads - The number of running threads for a given process. A thread is the basic unit of CPU utilization, and consists of a program counter, a register set, and a stack space. A thread is also called a lightweight process.
- Physical Memory Used - The amount of physical memory (or RAM) in kilobytes used by the specified process.
- Virtual Memory Used - The amount of virtual memory in kilobytes used by the specified process, or the size of the process in real memory plus swap.

Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. If no command name or PID is entered, the error Command not found is displayed and the probe will be set to a CRITICAL state.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.24. Linux::Process Health settings

Field	Value
Command Name	
Process ID (PID) file	
Timeout*	15
Critical Maximum CPU Usage	
Warning Maximum CPU Usage	
Critical Maximum Child Process Groups	
Warning Maximum Child Process Groups	
Critical Maximum Threads	
Warning Maximum Threads	
Critical Maximum Physical Memory Used	
Warning Maximum Physical Memory Used	
Critical Maximum Virtual Memory Used	
Warning Maximum Virtual Memory Used	

A.5.11. Linux::Process Running

The Linux::Process Running probe verifies that the specified process is functioning properly. It counts either processes or process groups, depending on whether the **Count process groups** checkbox is selected.

By default, the checkbox is selected, thereby indicating that the probe should count the number of process group leaders independent of the number of children. This allows you, for example, to verify that two instances of the Apache web server are running regardless of the (dynamic) number of child processes. If it is not selected, the probe conducts a straightforward count of the number of processes (children and leaders) matching the specified process.

Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. If no command name or PID is entered, the error Command not found is displayed and the probe enters a CRITICAL state.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.25. Linux::Process Running settings

Field	Value
Command name	
PID file	
Count process groups	(checked)
Timeout*	15
Critical Maximum Number Running	
Critical Minimum Number Running	

A.5.12. Linux::Swap Usage

The Linux::Swap Usage probe monitors the swap partitions running on a system and reports the following metric:

- Swap Free - The percent of swap memory currently free.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.26. Linux::Swap Usage settings

Field	Value
Timeout*	15
Warning Minimum Swap Free	
Critical Minimum Swap Free	

A.5.13. Linux::TCP Connections by State

The Linux::TCP Connections by State probe identifies the total number of TCP connections, as well as the quantity of each in the following states:

- TIME_WAIT - The socket is waiting after close for remote shutdown transmission so it may handle packets still in the network.
- CLOSE_WAIT - The remote side has been shut down and is now waiting for the socket to close.
- FIN_WAIT - The socket is closed, and the connection is now shutting down.
- ESTABLISHED - The socket has a connection established.
- SYN_RCVD - The connection request has been received from the network.

This probe can be helpful in finding and isolating network traffic to specific IP addresses or examining network connections into the monitored system.

The filter parameters for the probe let you narrow the probe's scope. This probe uses the **`netstat -ant`** command to retrieve data. The **`Local IP address`** and **`Local port`** parameters use values in the **`Local Address`** column of the output; the **`Remote IP address`** and **`Remote port`** parameters use values in the **`Foreign Address`** column of the output for reporting.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.27. Linux::TCP Connections by State settings

Field	Value
Local IP address filter pattern list	
Local port number filter	
Remote IP address filter pattern list	
Remote port number filter	
Timeout*	15
Critical Maximum Total Connections	
Warning Maximum Total Connections	
Critical Maximum TIME_WAIT Connections	
Warning Maximum TIME_WAIT Connections	
Critical Maximum CLOSE_WAIT Connections	
Warning Maximum CLOSE_WAIT Connections	
Critical Maximum FIN_WAIT Connections	
Warning Maximum FIN_WAIT Connections	
Critical Maximum ESTABLISHED Connections	
Warning Maximum ESTABLISHED Connections	
Critical Maximum SYN_RCVD Connections	
Warning Maximum SYN_RCVD Connections	

A.5.14. Linux::Users

The Linux::Users probe monitors the users of a system and reports the following metric:

- Users - The number of users currently logged in.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.28. Linux::Users settings

Field	Value
Timeout*	15
Critical Maximum Users	
Warning Maximum Users	

A.5.15. Linux::Virtual Memory

The Linux::Virtual Memory probe monitors the total system memory and collects the following metric:

- Virtual Memory - The percent of total system memory - random access memory (RAM) plus swap - that is free.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.29. Linux::Virtual Memory settings

Field	Value
Timeout*	15
Warning Minimum Virtual Memory Free	
Critical Minimum Virtual Memory Free	

A.6. LogAgent

The probes in this section monitor the log files on your systems. You can use them to query logs for certain expressions and track the sizes of files. For LogAgent probes to run, the **`nocpulse`** user must be granted read access to your log files.

Note that data from the first run of these probes is not measured against the thresholds to prevent spurious notifications caused by incomplete metric data. Measurements will begin on the second run.

A.6.1. LogAgent::Log Pattern Match

The LogAgent::Log Pattern Match probe uses regular expressions to match text located within the monitored log file and collects the following metrics:

- Regular Expression Matches - The number of matches that have occurred since the probe last ran.
- Regular Expression Match Rate - The number of matches per minute since the probe last ran.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe. For this probe to run, the **`nocpulse`** user must be granted read access to your log files.

In addition to the name and location of the log file to be monitored, you must provide a regular expression to be matched against. The expression must be formatted for **`egrep`**, which is equivalent to **`grep -E`** and supports extended regular expressions. This is the regular expression set for **`egrep`**:

```
^ beginning of line
$ end of line
. match one char
* match zero or more chars
[] match one character set, e.g. '[Ff]oo'
[^] match not in set '[^A-F]oo'
+ match one or more of preceding chars
? match zero or one of preceding chars
| or, e.g. a|b
() groups chars, e.g., (foo|bar) or (foo)+
```

Warning

Do not include single quotation marks (') within the expression. Doing so causes **egrep** to fail silently and the probe to time out.

Table A.30. LogAgent::Log Pattern Match settings

Field	Value
Log file*	/var/log/messages
Basic regular expression*	
Timeout*	45
Critical Maximum Matches	
Warning Maximum Matches	
Warning Minimum Matches	
Critical Minimum Matches	
Critical Maximum Match Rate	
Warning Maximum Match Rate	
Warning Minimum Match Rate	
Critical Maximum Match Rate	

A.6.2. LogAgent::Log Size

The LogAgent::Log Size probe monitors log file growth and collects the following metrics:

- Size - The size the log file has grown in bytes since the probe last ran.
- Output Rate - The number of bytes per minute the log file has grown since the probe last ran.
- Lines - The number of lines written to the log file since the probe last ran.
- Line Rate - The number of lines written per minute to the log file since the probe last ran.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe. For this probe to run, the **nocpulse** user must be granted read access to your log files.

Table A.31. LogAgent::Log Size settings

Field	Value
Log file*	/var/log/messages
Timeout*	20
Critical Maximum Size	
Warning Maximum Size	
Warning Minimum Size	
Critical Minimum Size	
Critical Maximum Output Rate	
Warning Maximum Output Rate	
Warning Minimum Output Rate	
Critical Minimum Output Rate	
Critical Maximum Lines	
Warning Maximum Lines	
Warning Minimum Lines	
Critical Minimum Lines	
Critical Maximum Line Rate	
Warning Maximum Line Rate	
Warning Minimum Line Rate	
Critical Minimum Line Rate	

A.7. MySQL 3.23 - 3.33

The probes in this section monitor aspects of the MySQL database using the **`mysqladmin`** binary. No specific user privileges are needed for these probes.

Note that the **`mysql-server`** package must be installed on the system conducting the monitoring for these probes to complete. See the MySQL Installation section of the *Red Hat Satellite Installation Guide* for instructions.

A.7.1. MySQL::Database Accessibility

The MySQL::Database Accessibility probe tests connectivity through a database account that has no database privileges. If no connection is made, a CRITICAL status results.

Table A.32. MySQL::Database Accessibility settings

Field	Value
Username*	
Password	
MySQL Port	3306
Database*	mysql
Timeout	15

A.7.2. MySQL::Opened Tables

The MySQL::Opened Tables probe monitors the MySQL server and collects the following metric:

- Opened Tables - The tables that have been opened since the server was started.

Table A.33. MySQL::Opened Tables settings

Field	Value
Username	
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Opened Objects	
Warning Maximum Opened Objects	
Warning Minimum Opened Objects	
Critical Minimum Opened Objects	

A.7.3. MySQL::Open Tables

The MySQL::Open Tables probe monitors the MySQL server and collects the following metric:

- Open Tables - The number of tables open when the probe runs.

Table A.34. MySQL::Open Tables settings

Field	Value
Username	
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Open Objects	
Warning Maximum Open Objects	
Warning Minimum Open Objects	
Critical Minimum Open Objects	

A.7.4. MySQL::Query Rate

The MySQL::Query Rate probe monitors the MySQL server and collects the following metric:

- Query Rate - The average number of queries per second per database server.

Table A.35. MySQL::Query Rate settings

Field	Value
Username	
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Query Rate	
Warning Maximum Query Rate	
Warning Minimum Query Rate	
Critical Minimum Query Rate	

A.7.5. MySQL::Threads Running

The MySQL::Threads Running probe monitors the MySQL server and collects the following metric:

- Threads Running - The total number of running threads within the database.

Table A.36. MySQL::Threads Running settings

Field	Value
Username	
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Threads Running	
Warning Maximum Threads Running	
Warning Minimum Threads Running	
Critical Minimum Threads Running	

A.8. Network Services

The probes in this section monitor various services integral to a functioning network. When applying them, ensure that their timed thresholds do not exceed the amount of time allotted to the timeout period. Otherwise, an UNKNOWN status is returned in all instances of extended latency, thereby nullifying the thresholds.

A.8.1. Network Services::DNS Lookup

The Network Services::DNS Lookup probe uses the **`dig`** command to see if it can resolve the system or domain name specified in the **`Host or Address to look up`** field. It collects the following metric:

- Query Time - The time in milliseconds required to execute the **`dig`** request.

This is useful in monitoring the status of your DNS servers. To monitor one of your DNS servers, supply a well-known host/domain name, such as a large search engine or corporate Web site.

Table A.37. Network Services::DNS Lookup settings

Field	Value
Host or Address to look up	
Timeout*	10
Critical Maximum Query Time	
Warning Maximum Query Time	

A.8.2. Network Services::FTP

The Network Services::FTP probe uses network sockets to test FTP port availability. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the FTP server to answer a connection request.

This probe supports authentication. Provide a username and password in the appropriate fields to use this feature.The optional **Expect** value is the string to be matched against after a successful connection is made to the FTP server. If the expected string is not found, the probe returns a CRITICAL state.

Table A.38. Network Services::FTP settings

Field	Value
Expect	FTP
Username	
Password	
FTP Port*	21
Timeout*	10
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.3. Network Services::IMAP Mail

The Network Services::IMAP Mail probe determines if it can connect to the IMAP 4 service on the system. Specifying an optional port will override the default port 143. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the IMAP server to answer a connection request.

The required **Expect** value is the string to be matched against after a successful connection is made to the IMAP server. If the expected string is not found, the probe returns a CRITICAL state.

Table A.39. Network Services::IMAP Mail settings

Field	Value
IMAP Port*	143
Expect*	OK
Timeout*	5
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.4. Network Services::Mail Transfer (SMTP)

The Network Services::Mail Transfer (SMTP) probe determines if it can connect to the SMTP port on the system. Specifying an optional port number overrides the default port 25. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the SMTP server to answer a connection request.

Table A.40. Network Services::Mail Transfer (SMTP) settings

Field	Value
SMTP Port*	25
Timeout*	10
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.5. Network Services::Ping

The Network Services::Ping probe determines if the Red Hat Satellite Server can **`ping`** the monitored system or a specified IP address. It also checks the packet loss and compares the round trip average against the Warning and Critical threshold levels. The required **`Packets to send`** value allows you to control how many ICMP ECHO packets are sent to the system. This probe collects the following metrics:

- Round-Trip Average - The time it takes in milliseconds for the ICMP ECHO packet to travel to and from the monitored system.
- Packet Loss - The percent of data lost in transit.

Although optional, the **`IP Address`** field can be instrumental in collecting metrics for systems that have multiple IP addresses. For instance, if the system is configured with multiple virtual IP addresses or uses Network Address Translation (NAT) to support internal and external IP addresses, this option may be used to check a secondary IP address rather than the primary address associated with the hostname.

Note that this probe conducts the **`ping`** from a Red Hat Satellite Server and not the monitored system. Populating the IP Address field does not test connectivity between the system and the specified IP address but between the Red Hat Satellite Server and the IP address. Therefore, entering the same IP address for Ping probes on different systems accomplishes precisely the same task. To conduct a **`ping`** from a monitored system to an individual IP address, use the Remote Ping probe instead. See Section A.8.7, “Network Services::Remote Ping”.

Table A.41. Network Services::Ping settings

Field	Value
IP Address (defaults to system IP)	
Packets to send*	20
Timeout*	10
Critical Maximum Round-Trip Average	
Warning Maximum Round-Trip Average	
Critical Maximum Packet Loss	
Warning Maximum Packet Loss	

A.8.6. Network Services::POP Mail

The Network Services::POP Mail probe determines if it can connect to the POP3 port on the system. A port number must be specified; specifying another port number overrides the default port 110. This probe collects the following metric:

- Remote Service Latency - The time it takes in seconds for the POP server to answer a connection request.

The required **`Expect`** value is the string to be matched against after a successful connection is made to the POP server. The probe looks for the string in the first line of the response from the system. The default is **`+OK`**. If the expected string is not found, the probe returns a CRITICAL state.

Table A.42. Network Services::POP Mail settings

Field	Value
Port*	110
Expect*	+OK
Timeout*	10
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.7. Network Services::Remote Ping

The Network Services::Remote Ping probe determines if the monitored system can **`ping`** a specified IP address. It also monitors the packet loss and compares the round trip average against the Warning and Critical threshold levels. The required **`Packets to send`** value allows you to control how many ICMP ECHO packets are sent to the address. This probe collects the following metrics:

- Round-Trip Average - The time it takes in milliseconds for the ICMP ECHO packet to travel to and from the IP address.
- Packet Loss - The percent of data lost in transit.

The **`IP Address`** field identifies the precise address to be pinged. Unlike the similar, optional field in the standard Ping probe, this field is required. The monitored system directs the ping to a third address, rather than to the Red Hat Satellite Server. Since the Remote Ping probe tests connectivity from the monitored system, another IP address must be specified. To conduct pings from the Red Hat Satellite Server to a system or IP address, use the standard Ping probe instead. See Section A.8.5, “Network Services::Ping”.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.43. Network Services::Remote Ping settings

Field	Value
IP Address*	
Packets to send*	20
Timeout*	10
Critical Maximum Round-Trip Average	
Warning Maximum Round-Trip Average	
Critical Maximum Packet Loss	
Warning Maximum Packet Loss	

A.8.8. Network Services::RPCService

The Network Services::RPCService probe tests the availability of remote procedure call (RPC) programs on a given IP address. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the RPC server to answer a connection request.

RPC server programs, which provide function calls via that RPC network, register themselves in the RPC network by declaring a program ID and a program name. NFS is an example of a service that works via the RPC mechanism.

Client programs that wish to use the resources of RPC server programs do so by asking the machine on which the server program resides to provide access to RPC functions within the RPC program number or program name. These conversations can occur over either TCP or UDP (but are almost always UDP).

This probe allows you to test simple program availability. You must specify the program name or number, the protocol over which the conversation occurs, and the usual timeout period.

Table A.44. Network Services::RPCService settings

Field	Value
Protocol (TCP/UDP)	udp
Service Name*	nfs
Timeout*	10
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.9. Network Services::Secure Web Server (HTTPS)

The Network Services::Secure Web Server (HTTPS) probe determines the availability of the secure Web server and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the HTTPS server to answer a connection request.

This probe confirms that it can connect to the HTTPS port on the specified host and retrieve the specified URL. If no URL is specified, the probe fetches the root document. The probe looks for a HTTP/1. message from the system unless you alter that value. Specifying another port number overrides the default port of 443.

This probe supports authentication. Provide a username and password in the appropriate fields to use this feature. Unlike most other probes, this probe returns a CRITICAL status if it cannot contact the system within the timeout period.

Table A.45. Network Services::Secure Web Server (HTTPS) settings

Field	Value
URL Path	/
Expect Header	HTTP/1
Expect Content	
UserAgent*	NOCpulse-check_http/1.0
Username	
Password	
Timeout*	10
HTTPS Port*	443
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.10. Network Services::SSH

The Network Services::SSH probe determines the availability of SSH on the specified port and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the SSH server to answer a connection request.

Upon successfully contacting the SSH server and receiving a valid response, the probe displays the protocol and server version information. If the probe receives an invalid response, it displays the message returned from the server and generates a WARNING state.

Table A.46. Network Services::SSH settings

Field	Value
SSH Port*	22
Timeout*	5
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.11. Network Services::Web Server (HTTP)

The Network Services::Web Server (HTTP) probe determines the availability of the Web server and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the HTTP server to answer a connection request.

This probe confirms it can connect to the HTTP port on the specified host and retrieve the specified URL. If no URL is specified, the probe will fetch the root document. The probe looks for a HTTP/1. message from the system, unless you alter that value. Specifying another port number will override the default port of 80. Unlike most other probes, this probe will return a CRITICAL status if it cannot contact the system within the timeout period.

This probe supports authentication. Provide a username and password in the appropriate fields to use this feature. Also, the optional Virtual Host field can be used to monitor a separate documentation set located on the same physical machine presented as a standalone server. If your Web server is not configured to use virtual hosts (which is typically the case), you should leave this field blank. If you do have virtual hosts configured, enter the domain name of the first host here. Add as many probes as necessary to monitor all virtual hosts on the machine.

Table A.47. Network Services::Web Server (HTTP) settings

Field	Value
URL Path	/
Virtual Host	
Expect Header	HTTP/1
Expect Content	
UserAgent*	NOCpulse-check_http/1.0
Username	
Password	
Timeout*	10
HTTP Port*	80
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.9. Oracle 8i, 9i, 10g, and 11g

The probes in this section may be applied to instances of the Oracle database matching the versions supported. Oracle probes require the configuration of the database and associations made by running the following command:

```
$ORACLE_HOME/rdbms/admin/catalog.sql
```

In addition, for these probes to function properly, the Oracle user configured in the probe must have minimum privileges of CONNECT and SELECT_CATALOG_ROLE.

Some Oracle probes are specifically aimed at tuning devices for long-term performance gains, rather than avoiding outages. Therefore, Red Hat recommends scheduling them to occur less frequently, between every hour and every two days. This provides a better statistical representation, de-emphasizing anomalies that can occur at shorter time intervals. This applies to following probes: Buffer Cache, Data Dictionary Cache, Disk Sort Ratio, Library Cache, and Redo Log.

For CRITICAL and WARNING thresholds based upon time to work as intended, their values cannot exceed the amount of time allotted to the timeout period. Otherwise, an UNKNOWN status is returned in all cases of extended latency, thereby nullifying the thresholds. For this reason, Red Hat strongly recommends ensuring that timeout periods exceed all timed thresholds. In this section, this refers specifically to the probe TNS Ping.

Finally, customers using these Oracle probes against a database using Oracle's Multi-Threaded Server (MTS) must contact Red Hat support to have entries added to the Red Hat Network Server's /etc/hosts file to ensure that the DNS name is resolved correctly.

A.9.1. Oracle::Active Sessions

The Oracle::Active Sessions probe monitors an Oracle instance and collects the following metrics:

- Active Sessions - The number of active sessions based on the value of **`V$PARAMETER.PROCESSES`**.
- Available Sessions - The percentage of active sessions that are available based on the value of **`V$PARAMETER.PROCESSES`**.

Table A.48. Oracle::Active Sessions settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Active Sessions	
Warning Maximum Active Sessions	
Critical Maximum Available Sessions Used	
Warning Maximum Available Sessions Used	

A.9.2. Oracle::Availability

The Oracle::Availability probe determines the availability of the database from the Red Hat Satellite.

Table A.49. Oracle::Availability settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30

A.9.3. Oracle::Blocking Sessions

The Oracle::Blocking Sessions probe monitors an Oracle instance and collects the following metric:

- Blocking Sessions - The number of sessions preventing other sessions from committing changes to the Oracle database, as determined by the required *Time Blocking* value you provide. Only those sessions that have been blocking for this duration, which is measured in seconds, are counted as blocking sessions.

Table A.50. Oracle::Blocking Sessions settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Time Blocking (seconds)*	20
Timeout*	30
Critical Maximum Blocking Sessions	
Warning Maximum Blocking Sessions	

A.9.4. Oracle::Buffer Cache

The Oracle::Buffer Cache probe computes the Buffer Cache Hit Ratio so as to optimize the system global area (SGA) Database Buffer Cache size. It collects the following metrics:

- Db Block Gets - The number of blocks accessed via single block gets (not through the consistent get mechanism).
- Consistent Gets - The number of accesses made to the block buffer to retrieve data in a consistent mode.
- Physical Reads - The cumulative number of blocks read from disk.
- Buffer Cache Hit Ratio - The rate at which the database goes to the buffer instead of the hard disk to retrieve data. A low ratio suggests more RAM should be added to the system.

Table A.51. Oracle::Buffer Cache settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port	1521
Timeout*	30
Warning Minimum Buffer Cache Hit Ratio	
Critical Minimum Buffer Cache Hit Ratio	

A.9.5. Oracle::Client Connectivity

The Oracle::Client Connectivity probe determines if the database is up and capable of receiving connections from the monitored system. This probe opens an **`rhnmd`** connection to the system and issues a **`sqlplus connect`** command on the monitored system.

The **`Expected DB name`** parameter is the expected value of **`V$DATABASE.NAME`**. This value is case-insensitive. A CRITICAL status is returned if this value is not found.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe. For this probe to run, the **`nocpulse`** user must be granted read access to your log files.

Table A.52. Oracle::Client Connectivity settings

Field	Value
Oracle Hostname or IP address*	
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
ORACLE_HOME*	/opt/oracle
Expected DB Name*	
Timeout*	30

A.9.6. Oracle::Data Dictionary Cache

The Oracle::Data Dictionary Cache probe computes the Data Dictionary Cache Hit Ratio so as to optimize the SHARED_POOL_SIZE in **`init.ora`**. It collects the following metrics:

- Data Dictionary Hit Ratio - The ratio of cache hits to cache lookup attempts in the data dictionary cache. In other words, the rate at which the database goes to the dictionary instead of the hard disk to retrieve data. A low ratio suggests more RAM should be added to the system.
- Gets - The number of blocks accessed via single block gets (not through the consistent get mechanism).
- Cache Misses - The number of accesses made to the block buffer to retrieve data in a consistent mode.

Table A.53. Oracle::Data Dictionary Cache settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Warning Minimum Data Dictionary Hit Ratio	
Critical Minimum Data Dictionary Hit Ratio	

A.9.7. Oracle::Disk Sort Ratio

The Oracle::Disk Sort Ratio probe monitors an Oracle database instance and collects the following metric:

- Disk Sort Ratio - The rate of Oracle sorts that were too large to be completed in memory and were instead sorted using a temporary segment.

Table A.54. Oracle::Disk Sort Ratio settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Disk Sort Ratio	
Warning Maximum Disk Sort Ratio	

A.9.8. Oracle::Idle Sessions

The Oracle::Idle Sessions probe monitors an Oracle instance and collects the following metric:

- Idle Sessions - The number of Oracle sessions that are idle, as determined by the required *Time Idle* value you provide. Only those sessions that have been idle for this duration, which is measured in seconds, are counted as idle sessions.

Table A.55. Oracle::Idle Sessions settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Time Idle (seconds)*	20
Timeout*	30
Critical Maximum Idle Sessions	
Warning Maximum Idle Sessions	

A.9.9. Oracle::Index Extents

The Oracle::Index Extents probe monitors an Oracle instance and collects the following metric:

- Allocated Extents - The number of allocated extents for any index.
- Available Extents - The percentage of available extents for any index.

The required **`Index Name`** field contains a default value of `%` that matches any index name.

Table A.56. Oracle::Index Extents settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Index Owner*	%
Index Name*	%
Timeout*	30
Critical Maximum of Allocated Extents	
Warning Maximum of Allocated Extents	
Critical Maximum of Available Extents	
Warning Maximum of Available Extents	

A.9.10. Oracle::Library Cache

The Oracle::Library Cache probe computes the Library Cache Miss Ratio so as to optimize the SHARED_POOL_SIZE in **`init.ora`**. It collects the following metrics:

- Library Cache Miss Ratio - The rate at which a library cache pin miss occurs. This happens when a session executes a statement that it has already parsed but finds that the statement is no longer in the shared pool.
- Executions - The number of times a pin was requested for objects of this namespace.
- Cache Misses - The number of pins that must now retrieve the object of the disk. These pins are made up of objects with previous pins from the time the object handle was created.

Table A.57. Oracle::Library Cache settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Library Cache Miss Ratio	
Warning Maximum Library Cache Miss Ratio	

A.9.11. Oracle::Locks

The Oracle::Locks probe monitors an Oracle database instance and collects the following metric:

- Active Locks - The current number of active locks as determined by the value in the v$locks table. Database administrators should be aware of high numbers of locks present in a database instance.

Locks are used so that multiple users or processes updating the same data in the database do not conflict. This probe is useful for alerting database administrators when a high number of locks are present in a given instance.

Table A.58. Oracle::Locks settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Active Locks	
Warning Maximum Active Locks	

A.9.12. Oracle::Redo Log

The Oracle::Redo Log probe monitors an Oracle database instance and collects the following metrics:

- Redo Log Space Request Rate - The average number of redo log space requests per minute since the server has been started.
- Redo Buffer Allocation Retry Rate - The average number of buffer allocation retries per minute since the server was started.

The metrics returned and the thresholds they are measured against are numbers representing the rate of change in events per minute. The rate of change for these metrics should be monitored because fast growth can indicate problems requiring investigation.

Table A.59. Oracle::Redo Log settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Redo Log Space Request Rate	
Warning Maximum Redo Log Space Request Rate	
Critical Maximum Redo Buffer Allocation Retry Rate	
Warning Maximum Redo Buffer Allocation Retry Rate	

A.9.13. Oracle::Table Extents

The Oracle::Table Extents probe monitors an Oracle database instance and collects the following metrics:

- Allocated Extents-Any Table - The total number of extents for any table.
- Available Extents-Any Table - The percentage of available extents for any table.

In Oracle, table extents allow a table to grow. When a table is full, it is *extended* by an amount of space configured when the table is created. Extents are configured on a per-table basis, with an extent size and a maximum number of extents.

For example, a table that starts with 10 MB of space and that is configured with an extent size of 1 MB and max extents of 10 can grow to a maximum of 20 MB (by being extended by 1 MB ten times). This probe can be configured to alert by (1) the number of allocated extents (e.g. "go critical when the table has been extended 5 or more times"), or (2) the table is extended past a certain percentage of its max extents (e.g. "go critical when the table has exhausted 80% or more of its max extents").

The required **`Table Owner`** and **`Table Name`** fields contain a default value of **`%`** that matches any table owner or name.

Table A.60. Oracle::Table Extents settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Table Owner*	%
Table Name*	%
Timeout*	30
Critical Maximum Allocated Extents	
Warning Maximum Allocated Extents	
Critical Maximum Available Extents	
Warning Maximum Available Extents	

A.9.14. Oracle::Tablespace Usage

The Oracle::Tablespace Usage probe monitors an Oracle database instance and collects the following metric:

- Available Space Used - The percentage of available space in each tablespace that has been used.

Tablespace is the shared pool of space in which a set of tables live. This probe alerts the user when the total amount of available space falls below the threshold. Tablespace is measured in bytes, so extents do not factor into it directly (though each extension removes available space from the shared pool).

The required **`Tablespace Name`** field is case insensitive and contains a default value of **`%`** that matches any table name.

Table A.61. Oracle::Tablespace Usage settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Tablespace Name*	%
Timeout*	30
Critical Maximum Available Space Used	
Warning Maximum Available Space Used	

A.9.15. Oracle::TNS Ping

The Oracle::TNS Ping probe determines if an Oracle listener is alive and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the Oracle server to answer a connection request.

Table A.62. Oracle::TNS Ping settings

Field	Value
TNS Listener Port*	1521
Timeout*	15
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.10. Red Hat Satellite

The probes in this section may be applied to the Red Hat Satellite itself to monitor its health and performance. Since these probes run locally, no specific application or transport protocols are required.

A.10.1. Red Hat Satellite::Disk Space

The Red Hat Satellite::Disk Space probe monitors the free disk space on a Satellite and collects the following metrics:

- File System Used - The percent of the current file system now in use.
- Space Used - The file size used by the current file system.
- Space Available - The file size available to the current file system.

Table A.63. Red Hat Satellite::Disk Space settings

Field	Value
Device Pathname*	/dev/hda1
Critical Maximum File System Used	
Warning Maximum File System Used	
Critical Maximum Space Used	
Warning Maximum Space Used	
Critical Maximum Space Available	
Warning Maximum Space Available	

A.10.2. Red Hat Satellite::Execution Time

The Red Hat Satellite::Execution Time probe monitors the execution time for probes run from a Satellite and collects the following metric:

- Probe Execution Time Average - The seconds required to fully execute a probe.

Table A.64. Red Hat Satellite::Execution Time settings

Field	Value
Critical Maximum Probe Execution Time Average	
Warning Maximum Probe Execution Time Average	

A.10.3. Red Hat Satellite::Interface Traffic

The Red Hat Satellite::Interface Traffic probe monitors the interface traffic on a Satellite and collects the following metrics:

- Input Rate - The amount of traffic in bytes per second the device receives.
- Output Rate - The amount of traffic in bytes per second the device sends.

Table A.65. Red Hat Satellite::Interface Traffic settings

Field	Value
Interface*	eth0
Timeout (seconds)*	30
Critical Maximum Input Rate	
Critical Maximum Output Rate	

A.10.4. Red Hat Satellite::Latency

The Red Hat Satellite::Latency probe monitors the latency of probes on a Satellite and collects the following metric:

- Probe Latency Average - The lag in seconds between the time a probe becomes ready to run and the time it is actually run. Under normal conditions, this is generally less than a second. When a Satellite is overloaded (because it has too many probes with respect to their average execution time), the number goes up.

Table A.66. Red Hat Satellite::Latency settings

Field	Value
Critical Maximum Probe Latency Average	
Warning Maximum Probe Latency Average	

A.10.5. Red Hat Satellite::Load

The Red Hat Satellite::Load probe monitors the CPU load on a Satellite and collects the following metric:

- Load - The load average on the CPU for a 1-, 5-, and 15-minute period.

Table A.67. Red Hat Satellite::Load settings

Field	Value
Critical Maximum 1-minute Average	
Warning Maximum 1-minute Average	
Critical Maximum 5-minute Average	
Warning Maximum 5-minute Average	
Critical Maximum 15-minute Average	
Warning Maximum 15-minute Average	

A.10.6. Red Hat Satellite::Probe Count

The Red Hat Satellite::Probe Count probe monitors the number of probes on a Satellite and collects the following metric:

- Probes - The number of individual probes running on a Satellite.

Table A.68. Red Hat Satellite::Probe Count settings

Field	Value
Critical Maximum Probe Count	
Warning Maximum Probe Count	

A.10.7. Red Hat Satellite::Process Counts

The Red Hat Satellite::Process Counts probe monitors the number of processes on a Satellite and collects the following metrics:

- Blocked - The number of processes that have been switched to the waiting queue and waiting state.
- Child - The number of processes spawned by another process already running on the machine.
- Defunct - The number of processes that have terminated (either because they have been killed by a signal or have called **`exit()`**) and whose parent processes have not yet received notification of their termination by executing some form of the **`wait()`** system call.
- Stopped - The number of processes that have stopped before their executions could be completed.
- Sleeping - A process that is in the **`Interruptible`** sleep state and that can later be reintroduced into memory, resuming execution where it left off.

Table A.69. Red Hat Satellite::Process Counts settings

Field	Value
Critical Maximum Blocked Processes	
Warning Maximum Blocked Processes	
Critical Maximum Child Processes	
Warning Maximum Child Processes	
Critical Maximum Defunct Processes	
Warning Maximum Defunct Processes	
Critical Maximum Stopped Processes	
Warning Maximum Stopped Processes	
Critical Maximum Sleeping Processes	
Warning Maximum Sleeping Processes	

A.10.8. Red Hat Satellite::Processes

The Red Hat Satellite::Processes probe monitors the number of processes on a Satellite and collects the following metric:

- Processes - The number of processes running simultaneously on the machine.

Table A.70. Red Hat Satellite::Processes settings

Field	Value
Critical Maximum Processes	
Warning Maximum Processes	

A.10.9. Red Hat Satellite::Process Health

The Red Hat Satellite::Process Health probe monitors customer-specified processes and collects the following metrics:

- CPU Usage - The CPU usage percent for a given process.
- Child Process Groups - The number of child processes spawned from the specified parent process. A child process inherits most of its attributes, such as open files, from its parent.
- Threads - The number of running threads for a given process. A thread is the basic unit of CPU utilization, and consists of a program counter, a register set, and a stack space. A thread is also called a lightweight process.
- Physical Memory Used - The amount of physical memory in kilobytes being used by the specified process.
- Virtual Memory Used - The amount of virtual memory in kilobytes being used by the specified process, or the size of the process in real memory plus swap.

Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. If no command name or PID is entered, the error Command not found is displayed and the probe is set to a CRITICAL state.

Table A.71. Red Hat Satellite::Process Health settings

Field	Value
Command Name	
Process ID (PID) file	
Timeout*	15
Critical Maximum CPU Usage	
Warning Maximum CPU Usage	
Critical Maximum Child Process Groups	
Warning Maximum Child Process Groups	
Critical Maximum Threads	
Warning Maximum Threads	
Critical Maximum Physical Memory Used	
Warning Maximum Physical Memory Used	
Critical Maximum Virtual Memory Used	
Warning Maximum Virtual Memory Used	

A.10.10. Red Hat Satellite::Process Running

The Red Hat Satellite::Process Running probe verifies that the specified process is running. Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. A Critical status results if the probe cannot verify the command or PID.

Table A.72. Red Hat Satellite::Process Running settings

Field	Value
Command Name	
Process ID (PID) file	
Critical Number Running Maximum	
Critical Number Running Minimum	

A.10.11. Red Hat Satellite::Swap

The Red Hat Satellite::Swap probe monitors the percent of free swap space available on a Satellite. A CRITICAL status results if the value falls below the Critical threshold. A WARNING status results if the value falls below the Warning threshold.

Table A.73. Red Hat Satellite::Swap settings

Field	Value
Critical Minimum Swap Percent Free	
Warning Minimum Swap Percent Free	

A.10.12. Red Hat Satellite::Users

The Red Hat Satellite::Users probe monitors the number of users currently logged into a Satellite. A CRITICAL status results if the value exceeds the Critical threshold. A WARNING status results if the

value exceeds the Warning threshold.

Table A.74. Red Hat Satellite::Users settings

Field	Value
Critical Maximum Users	
Warning Maximum Users	

Revision History

Revision 4-33.400 **2013-10-31** **Rüdiger Landmann**
Rebuild with publican 4.0.0

Revision 4-33 **Fri Sep 27 2013** **Dan Macpherson**
Final version of documentation suite

Revision 4-32 **Thu Aug 29 2013** **Dan Macpherson**
First implementation of QE Review feedback

Revision 4-31 **Tue Aug 27 2013** **Dan Macpherson**
Minor changes

Revision 4-30 **Tue Aug 27 2013** **Dan Macpherson**
Final QE implementation

Revision 4-29 **Tue Aug 27 2013** **Dan Macpherson**
Fixing screen text

Revision 4-28 **Tue Aug 27 2013** **Dan Macpherson**
Removing computeroutput tags

Revision 4-27 **Tue Aug 27 2013** **Dan Macpherson**
Implementation of feedback from BZ#1001385

Revision 4-26 **Tue Aug 27 2013** **Dan Macpherson**
Implementing QE feedback from BZ#1001385

Revision 4-25 **Tue Aug 27 2013** **Dan Macpherson**
Minor typo fix for BZ#1001378

Revision 4-24 **Tue Aug 27 2013** **Dan Macpherson**
Implementation of QE feedback based on BZ#1001378 and BZ#1001379

Revision 4-23 **Tue Aug 27 2013** **Dan Macpherson**
QE Feedback Implementation for BZ#1001376

Revision 4-22 **Thu Aug 15 2013** **Dan Macpherson**
Typo corrections from QE Review

Revision 4-21 **Sun Jul 28 2013** **Dan Macpherson**
Second implementation of tech review feedback

Revision 4-20 **Wed Jul 24 2013** **Dan Macpherson**
Corrections for BZ#987245

Revision 4-19 **Tue Jul 23 2013** **Dan Macpherson**
First implementation of tech review feedback

Revision 4-18 **Thu Jul 12 2013** **Dan Macpherson**
Final beta updates

Revision 4-17 **Thu Jul 12 2013** **Dan Macpherson**
Beta update

Revision 4-16 **Thu Jul 11 2013** **Athene Chan**
Edited Splice section.
Additional ISS content added.

Revision 4-15 **Fri Jul 5 2013** **Athene Chan**
BZ#906577 ISS editing from developer reviews.

Revision 4-14 **Fri Jul 5 2013** **Athene Chan**
BZ#906577 Additional information about ISS new features have been included.

Revision 4-13 **Fri June 28 2013** **Athene Chan**
Updated all sections based on updated UI changes.
Changed all "Red Hat Proxy" to "Red Hat Satellite Proxy" based on branding name change.
BZ#906577 Added Intersatellite-sync information to book.

Revision 4-12 **Tue June 4 2013** **Athene Chan**
BZ#969091 Changed outdated filename from /etc/rhn/rhn_web.conf to /etc/rhn/rhn.conf.

Revision 4-11 **Fri May 17 2013** **Athene Chan**
Edited book procedures based on the user interface.
Staged for review.

Revision 4-10 **Thu Apr 25 2013** **Athene Chan**
BZ#908911 All up2date references have been changed to current versions.
BZ#927113, 950295 Book abstract has been updated
BZ#927546, 924221 Minor edits to standardized terms
Edited content for the next version release.

Revision 4-9 **Thu Feb 28 2013** **Athene Chan**
Edited Table of Contents in preparation for the next version release.

Revision 4-8 **Wed Jan 2 2013** **Athene Chan**
BZ#862950 Space between "(pup)" and "that" included.

Revision 4-7 **Wed Sept 19 2012** **Dan Macpherson**
Final packaging for 5.5

Revision 4-6 **Thu Aug 16 2012** **Athene Chan**
BZ#847993 Changed filename on example in section 5.2.4

Revision 4-5 **Thu Aug 16 2012** **Athene Chan**
BZ#773647 updated paragraphs pertaining to the "create new account" screenshot
BZ#846691 updated "buy" link in Section 1.1

Revision 4-4 **Wed Aug 15 2012** **Athene Chan**
BZ#773647 updated "create new account" screenshot

Revision 4-3 **Thu Aug 9 2012** **Athene Chan**
Staging documents for review

Revision 3-2 **Fri Aug 3 2012** **Athene Chan**
BZ#844849 Restructured paragraph.

Revision 3-1 **Tue Jun 17 2012** **Athene Chan**
Deprecated content removed. Prepared for 5.5 Release
BZ#837703 Custom GPG Key note added

Revision 3-0 **Thurs May 24 2012** **Athene Chan**
BZ#783340 - Updated "s390x" to "IBM System z"

Revision 2-6 **Mon Jan 9 2012** **Lana Brindley**
BZ#707591 - Virtualization chapter - update instructions
BZ#746640 - Virtualization chapter - added kickstart information

Revision 2-5 **Wed Jan 4 2012** **Lana Brindley**
BZ#707568 & BZ#707570 - Virtualization chapter - channel names
BZ#744653 - Virtualization chapter - typos
BZ#744656 - Virtualization chapter - update RHEL6 instructions
BZ#750481 - Updated method for changing max file size
BZ#766424 - Kickstart chapter - updated text

Revision 2-4 **Fri Sep 23 2011** **Lana Brindley**
BZ#702516 - Unix handbook
BZ#703605 - Monitoring chapter
BZ#706868 & BZ#707169 - Cobbler chapter
BZ#707591 - Virtualization chapter
BZ#707602 - Virtualization chapter
BZ#715267 - Typos

Revision 2-3 **Mon Aug 15 2011** **Lana Brindley**
Folded z-stream release into y-stream

Revision 2-2 **Wed Jun 15 2011** **Lana Brindley**
Prepared for publication

Revision 2-1 **Fri May 27 2011** **Lana Brindley**
Updates from translators

Revision 2-0 **Fri May 6 2011** **Lana Brindley**
Prepared for translation

Revision 1-29 **Fri March 25 2011** **Lana Brindley**
Fixed entities for translation

BZ#683466 - Monitoring

Revision 1-28 **Thu March 24 2011** **Lana Brindley**

BZ#679621 - Fix entities for translation
BZ#681788 - Notifications

Revision 1-27 **Mon Feb 14 2011** **Lana Brindley**

BZ#658127 - API Access

Revision 1-26 **Wed Feb 9 2011** **Lana Brindley**

BZ#658120 - Remove RHEL 2.1 references
BZ#658131 - API Access
BZ#669166 - Virtualization

Revision 1-25 **Mon Jan 31 2011** **Lana Brindley**

BZ#443630 - Kickstart
BZ#559515 - Cobbler